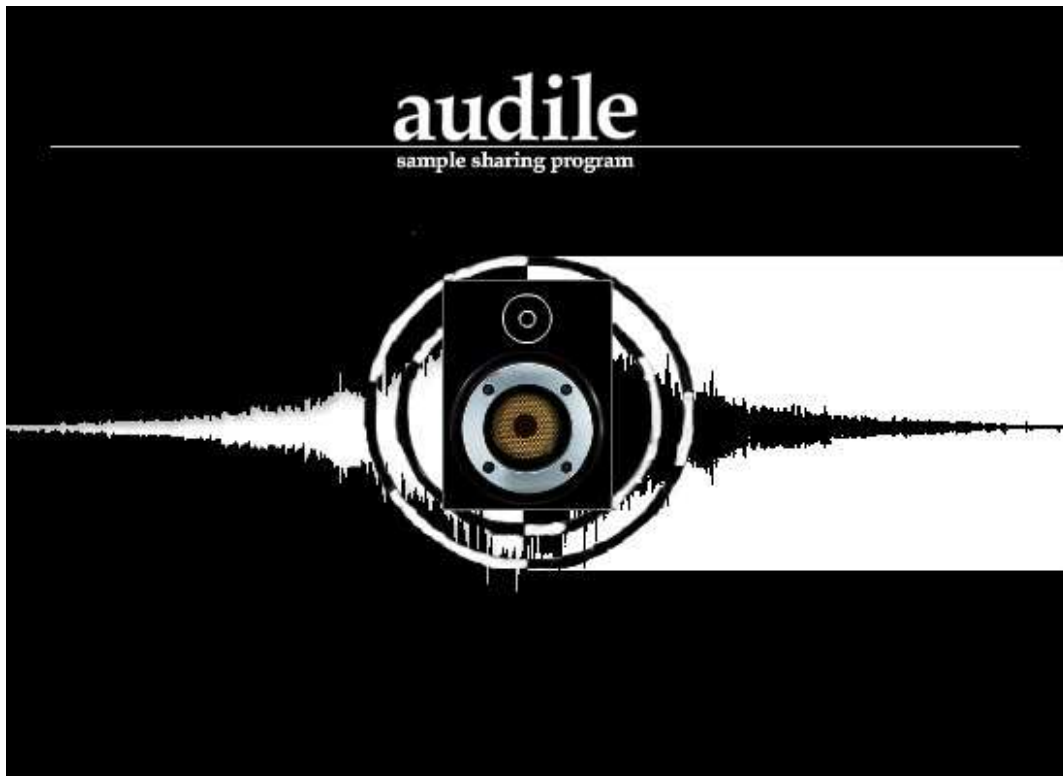


ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ  
ΠΑΡΑΡΤΗΜΑ ΡΕΘΥΜΝΟΥ  
ΤΜΗΜΑ ΜΟΥΣΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ & ΑΚΟΥΣΤΙΚΗΣ

## Πτυχιακή Εργασία

**Audile: Δικτυακή πύλη για την ανταλλαγή ηχητικών  
δειγμάτων μέσω εξειδικευμένης βάσης δεδομένων**



**Σπουδαστής: Γιώργος Λαμπρόπουλος (Α.Μ 126)**

**Επιβλέπουσα: Χρυσούλα Αλεξανδράκη**

**Ρέθυμνο 2007**

# Πίνακας Περιεχομένων

1	Εισαγωγή.....	4
2	Δικτυακά συστήματα ανταλλαγής ηχητικού περιεχομένου .....	5
3	Σχεδιασμός συστήματος.....	6
3.1	Λειτουργικότητα .....	6
3.2	Σχεδιασμός Βάσης Δεδομένων.....	7
3.3	Σχεδιασμός διεπαφής χρήστη (user interface) .....	8
4	Υλοποίηση .....	12
4.1	Αρχιτεκτονική στην οποία βασίζεται ο δικτυακός τόπος audile.....	12
4.2	Εργαλεία Ανάπτυξης .....	12
4.2.1	HTML .....	13
4.2.2	PHP .....	14
4.2.2.1	Ιστορικά .....	14
4.2.2.2	Λειτουργία.....	14
4.2.2.3	Τι είναι η PHP .....	15
4.2.2.4	Τι μπορεί να κάνει η PHP;.....	16
4.2.3	Ηλεκτρονικές βάσεις δεδομένων .....	18
4.2.3.1	MySQL.....	19
4.2.3.2	Επιδόσεις .....	20
4.2.4	APACHE.....	21
4.3	Ανταλλαγή ηχητικού περιεχομένου .....	22
4.4	Πλοήγηση στο πρόγραμμα .....	26
4.5	AudileRadio .....	44
4.5.1	Streaming Media .....	44
4.5.2	Αποθήκευση Streaming και Bandwidth .....	46
4.5.3	Πρωτόκολλα Streaming Media.....	46

4.5.4	Shoutcast .....	49
5	Συμπεράσματα.....	52
6	Αναφορές .....	53
6.1	Βιβλιογραφία .....	53
6.2	Διαδικτυακές Παραπομπές .....	53

# 1 Εισαγωγή

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη μίας δικτυακής πύλης (portal) η οποία επιτρέπει την πρόσβαση σε ηχητικό υλικό και την ανταλλαγή αρχείων ήχου. Ανώτερος στόχος είναι η διάθεση της εφαρμογής αυτής στους σπουδαστές του Τμήματος Μουσικής Τεχνολογίας και Ακουστικής, ώστε να χρησιμεύσει ως εργαλείο για την αποθήκευση και την προβολή του ηχητικού υλικού που παράγουν στα πλαίσια των σπουδών και των ενδιαφερόντων τους. Η υλοποίηση της εφαρμογής είναι βασισμένη στη χρήση σύγχρονων τεχνολογιών Διαδικτύου, και είναι έτοιμη για να διατεθεί στους σπουδαστές, μαζί με τη νέα ιστοσελίδα του Τμήματος.

Η δικτυακή πύλη που έχει αναπτυχθεί στα πλαίσια αυτής της πτυχιακής εργασίας ονομάζεται Audile. Ο αγγλικός αυτός όρος σημαίνει ακουστικός, ο έχον ακουστική φαντασία. Πιο συγκεκριμένα, η δικτυακή πύλη Audile, διαθέτει μια βάση δεδομένων η οποία δίνει εξατομικευμένη πρόσβαση σε αρχεία ήχου που είναι ταξινομημένα ανά χρήστη. Τα αρχεία αυτά μπορούν να είναι είτε δείγματα ήχου (samples) είτε ολοκληρωμένες μίξεις και ηχογραφήσεις των χρηστών του συστήματος. Επιπλέον πέρα από τη βάση δεδομένων το Audile παρέχει και ένα δικτυακό ραδιόφωνο εύκολα προσβάσιμο από όλους, όπου θα μπορούν να μεταδίδονται συνεντεύξεις, νέα καθώς και κανονικές ζωντανές εκπομπές.

Σκοπός του Audile είναι αυτό ουσιαστικά που μας δηλώνει με το όνομά του. Να μας επηρεάσει και να μας βοηθήσει να αποκτήσουμε μια ακουστική φαντασία που θα μας είναι χρήσιμη και για τα χρόνια που θα βρισκόμαστε μέσα στο τμήμα, στις διάφορες μίξεις μας, αλλά και για τα χρόνια που θα έχουμε πλέον φύγει από το τμήμα όπου θα ξέρουμε πως μόνο και μόνο σαν απόφοιτοι θα έχουμε πρόσβαση σε μια τράπεζα ήχων. Πέρα από την ιδιότητά του σαν sample sharing πρόγραμμα το ραδιόφωνο του audile έχει σαν βασικό σκοπό να ενισχύσει την μαθητική κοινότητα του τμήματος μουσικής τεχνολογίας και ακουστικής και μέσω της εκτεταμένης χρήσης του να βοηθήσει στην ανάδειξη κάποιων φοιτητών που ενδιαφέρονται για την μουσική σύνθεση και παραγωγή. Επιπλέον μπορεί να δώσει κύρος και ισχύ στο τμήμα προκειμένου να ακουστούν πιθανά προβλήματα και ελλείψεις καθώς επίσης να βοηθήσει πολιτιστικά την πόλη του Ρεθύμνου, με ένα πρωτοποριακό ραδιόφωνο κάτι που στην ουσία δεν υπάρχει. Τελικός σκοπός του Audile είναι η επιπλέον ανάδειξη της ιστοσελίδας του τμήματος μουσικής τεχνολογίας και ακουστικής, όπου θα υπάρχει σχετικό link, η οποία ανακατασκευάζεται με σκοπό την αύξηση της λειτουργικότητας καθώς και της επισκεψιμότητάς της.

Το παρόν κείμενο έχει δομηθεί ως εξής: Στο δεύτερο κεφάλαιο, που έχει τον τίτλο «Δικτυακά συστήματα ανταλλαγής ηχητικού περιεχομένου» παρουσιάζονται άλλα, ήδη υπάρχοντα συστήματα ανταλλαγής ηχητικού περιεχομένου, τα οποία διατίθενται στο Διαδίκτυο. Τα κεφάλαια που ακολουθούν περιγράφουν τα στάδια που ακολουθήθηκαν κατά την ανάπτυξη του Audile, σχεδιασμός υλοποίηση και αξιολόγηση, ενώ τέλος παρατίθενται κάποια συμπεράσματα σχετικά με την χρηστικότητα και την προσβασιμότητα της δικτυακής αυτής πύλης.

## 2 Δικτυακά συστήματα ανταλλαγής ηχητικού περιεχομένου

Στο κεφάλαιο αυτό αναφέρονται ενδεικτικά κάποιες δικτυακές πύλες ανταλλαγής ηχητικού περιεχομένου, οι οποίες διατίθενται στο Διαδίκτυο.

Μια τέτοια πύλη είναι το Freesound [3] που επιτρέπει την πρόσβαση σε μια πληθώρα, μουσικών και μη, ηχητικών δειγμάτων. Κατασκευάστρια εταιρία είναι η Creative Commons η οποία έχει και τα δικαιώματα του συγκεκριμένου project (Creative Commons Sampling Plus License). Ο αριθμός των ηχητικών δειγμάτων που διατίθενται μέσω της ιστοσελίδας του freesound αυτή την περίοδο είναι περί τα 20000 δείγματα. Αντιλαμβανόμαστε λοιπόν πώς ο προγραμματισμός ιστοσελίδων με σκοπό την ανταλλαγή δειγμάτων είναι πολύ πρόσφατος μιας και τα 20000 δείγματα δεν είναι κάποιος πολύ μεγάλος αριθμός και το freesound είναι το πιο διαδεδομένο site που κάνει αυτή την δουλειά. Η αλήθεια είναι πως ο αριθμός αυτών των προγραμμάτων είναι περιορισμένος στο internet.

Άλλο ένα sample sharing site είναι το findsounds.com [4]. Σε αυτό τα πράγματα είναι πιο άπλα αφού στην πρώτη σελίδα του site υπάρχει μία λειτουργία αναζήτησης (search) και από αυτό ο χρήστης μπορεί να βρει τους ήχους που αναζητά. Επίσης του δίνεται η δυνατότητα να ορίσει κάποια χαρακτηριστικά του ήχου όπως το format, τον αριθμό των καναλιών, το ελάχιστο επιθυμητό bit resolution και sample rate καθώς και το μέγιστο όγκο του αρχείου.

Το findsounds.com αν και λιγότερο διαδεδομένο είναι μάλλον πιο εύχρηστο και λειτουργικό από το freesound.com το οποίο από την άλλη δίνει την εντύπωση στον χρήστη ότι ανήκει σε κάποια ομάδα ανθρώπων με κοινά ενδιαφέροντα καθώς παρέχει πληροφορίες όπως ο χρήστης της εβδομάδας ή το ποιος είναι ο πιο δραστήριος χρήστης με τα πιο πολλά αρχεία.

Βέβαια σύντομα θα υπάρχουν πολλά τέτοιου είδους προγράμματα μιας και η ανταλλαγή δειγμάτων ήχου σύντομα θα αποδειχθεί πολλή χρήσιμη καθώς και πιθανότατα επικερδής. Ήδη πολλοί προγραμματιστές κατευθύνονται σε αυτόν τον τομέα και γενικότερα σε ότι αφορά το audio και video streaming.

## 3 Σχεδιασμός συστήματος

### 3.1 Λειτουργικότητα

- **Προσωπικός λογαριασμός εισόδου για κάθε χρήστη**

Για να κάνει κανείς login στο Audile πρέπει τα αρχικά του αλλά και ο αριθμός μητρώου του να ταιριάζουν με αυτά που δίνονται στην φόρμα της πρώτης σελίδας που εμφανίζεται. Τα δεδομένα αναγνώρισης του χρήστη υπάρχουν αποθηκευμένα στην βάση δεδομένων και εάν ταυτίζονται με αυτά που δίνει ο χρήστης τότε του επιτρέπεται και η είσοδος στο Audile. Με αυτό τον τρόπο επιτυγχάνεται η χρήση του Audile να είναι αποκλειστικά και μόνο για τους σπουδαστές του τμήματος ως προς τις λειτουργίες ανταλλαγής δειγμάτων κάτι που διευκολύνει τον εξυπηρετητή του Audile. Επίσης με αυτόν τον τρόπο μειώνεται και ο αποθηκευτικός χώρος που απαιτείται στον εξυπηρετητή.

- **Λειτουργίες Download και Upload για συγκεκριμένα audio formats**

Τα δείγματα που επιτρέπει το Audile να φορτωθούν στην βάση δεδομένων του είναι δείγματα συγκεκριμένων format. Το format αυτό μπορεί να είναι wav, aiff, mp3 και ogg. Τα παραπάνω format αν εξαιρέσουμε το wma είναι τα πιο ευρέως διαδεδομένα format που χρησιμοποιούνται για την κωδικοποίηση του ήχου.

- **Καταγραφή χαρακτηριστικών των δειγμάτων στη βάση δεδομένων**

Όλα τα χαρακτηριστικά ενός δείγματος καταγράφονται στη βάση δεδομένων στον πίνακα files. Τα χαρακτηριστικά αυτά είναι το όνομα, το μέγεθος, τα bits, η ημερομηνία, η διεύθυνση του στον εξυπηρετητή, ο αριθμός καναλιών, το format, η δειγματοληψία, το μήκος του και ο τύπος συμπίεσης. Έτσι μπορεί ο χρήστης να συγκρίνει τα αρχεία και να αποφασίσει πιο προτιμάει ή πιο του κάνει για την δουλειά που το θέλει. Υπεύθυνο για αυτήν την λειτουργία είναι το `AudioClass.php` αρχείο, το οποίο έχει προγραμματιστεί ώστε να μπορεί να αποκωδικοποιεί αυτά τα χαρακτηριστικά από τα συγκεκριμένα format ηχητικών δειγμάτων.

- **Αυτόματη απεικόνιση κυματομορφής**

Όταν ο χρήστης επιλέξει να δει τις πληροφορίες ενός δείγματος τότε το `AudioClass` αρχείο σπεύδει να κοιτάξει τον τύπο του αρχείου. Στην περίπτωση που είναι wav ή aiff, το Audile είναι σε θέση να απεικονίσει την κυματομορφή του αρχείου.

- **Καταγραφή μεταδεδομένων που αφορούν το ηχητικό δείγμα**

Ένας τύπος μετα-δεδομένων είναι και τα σχόλια που μας επιτρέπει το Audile να φορτώσει ο χρήστης καθώς ανεβάζει ένα δείγμα στην βάση του προγράμματος. Αυτά τα σχόλια μπορούν να βοηθήσουν πολύ στην αναζήτηση ενός δείγματος καθώς το πρόγραμμα ψάχνει και τα σχόλια για πιθανές λέξεις κλειδιά καθώς και τα ονόματα των αρχείων.

- **Internet radio**

Τέλος το Audile παρέχει σε όλους τους χρήστες ανεξαιρέτως του προσωπικού τους λογαριασμού, να ακούσουν ζωντανά το ραδιόφωνο που φιλοξενεί. Το ραδιόφωνο εκπέμπει μέσω του Shoutcast που είναι ένας εξυπηρετητής αποκλειστικά για αυτήν την χρήση.

### 3.2 Σχεδιασμός Βάσης Δεδομένων

Παρακάτω παρατηρούμε πως έχει δομηθεί η βάση δεδομένων του Audile.

Η βάση περιέχει 3 πίνακες όπως φαίνεται και στον πίνακα 3.1. Οι πίνακες αυτοί ονομαστικά είναι: school\_users, directories, files.

school_users	directories	files
<u>user_id</u> (1) primary	<u>dir_id</u> (1) primary	<u>file_id</u> (1) primary
<u>user_am</u> (1) primary	user_jd ∞ secondary	filename
email	dir_name	filesize
initials	dir_type	bitrate
name		date
surname		url
no_logins		resolution
session_id		no_channels
		format
		sampling_rate
		dir_id ∞ secondary
		length
		comments
		compression
		type

**Πίνακας 3.1: Database Schema**

Οι σειρές και οι στήλες αυτών όπως καταχωρούνται στην βάση δεδομένων παρουσιάζονται στους πίνακες από 3.2 μέχρι 3.4 και είναι οι εξής:

	Field	Type	Attributes	Null	Default	Extra	Action						
<input type="checkbox"/>	<u>user_id</u>	int(11)		No		auto_increment							
<input type="checkbox"/>	<u>user_am</u>	varchar(10)		No									
<input type="checkbox"/>	email	varchar(25)		No									
<input type="checkbox"/>	initials	varchar(5)		No									
<input type="checkbox"/>	name	varchar(25)		No									
<input type="checkbox"/>	surname	varchar(100)		No									
<input type="checkbox"/>	no_logins	int(100)		No	0								
<input type="checkbox"/>	session_id	varchar(255)		No									

**Πίνακας 3.2: school\_users**

	Field	Type	Attributes	Null	Default	Extra	Action					
<input type="checkbox"/>	<u>dir_id</u>	int(5)		No		auto_increment						
<input type="checkbox"/>	user_id	int(5)		No	0							
<input type="checkbox"/>	dir_name	varchar(30)		No								
<input type="checkbox"/>	dir_type	varchar(50)		No								

**Πίνακας 3.3: directories**

	Field	Type	Attributes	Null	Default	Extra	Action					
<input type="checkbox"/>	<u>file_id</u>	int(10)	UNSIGNED	No		auto_increment						
<input type="checkbox"/>	filename	varchar(255)		No								
<input type="checkbox"/>	filesize	varchar(255)		No								
<input type="checkbox"/>	bitrate	varchar(255)		No								
<input type="checkbox"/>	date	date		No	0000-00-00							
<input type="checkbox"/>	url	varchar(255)		No								
<input type="checkbox"/>	resolution	varchar(255)		No								
<input type="checkbox"/>	no_channels	varchar(255)		No								
<input type="checkbox"/>	format	varchar(255)		No								
<input type="checkbox"/>	sampling_rate	varchar(255)		No								
<input type="checkbox"/>	dir_id	int(5)		No	0							
<input type="checkbox"/>	length	varchar(100)		No								
<input type="checkbox"/>	comments	varchar(100)		No								
<input type="checkbox"/>	compression	varchar(255)		Yes	NULL							
<input type="checkbox"/>	type	varchar(20)		No								

**Πίνακας 3.4: files**

Ο πίνακας school\_users έχει ως πρωτεύοντα ορίσματα τα πεδία user\_am και user\_id. Όταν αυτά είναι σωστά σε σχέση με τα στοιχεία που δίνει ο χρήστης στο login τότε ο χρήστης μπορεί να έχει πρόσβαση στον πίνακα directories με πρωτεύων όρισμα το πεδίο dir\_id. Με βάση το dir\_id στην συνέχεια ο χρήστης μπορεί να αναζητήσει είτε να ανεβάσει αρχεία στον πίνακα files με πρωτεύων όρισμα το file\_id. Δηλαδή όλα είναι συσχετισμένα κάτω από ένα user\_id και user\_am σε όλα τα tables.

Με αυτόν τον τρόπο αυτό που καταφέρνουμε είναι να έχουμε μια ευανάγνωστη, εύχρηστη και λειτουργική βάση δεδομένων όπου η οποιαδήποτε αναζήτηση μας είναι εύκολο να πραγματοποιηθεί και να μας δώσει τα σωστά αποτελέσματα.

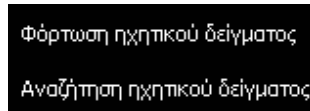
### 3.3 Σχεδιασμός διεπαφής χρήστη (user interface)

Μετά από το εισαγωγικό animation του Audile ο χρήστης βλέπει την κυρίως σελίδα. Εκεί του ζητούνται τα στοιχεία του που είναι ο αριθμός μητρώου και τα αρχικά του. Ακόμα δίνονται και κάποιες οδηγίες χρήσης μαζί με το εισαγωγικό σημείωμα. Άμα είναι η πρώτη φορά που εισέρχεται στον δικτυακό τόπο του Audile τότε πριν περάσει στην ανταλλαγή δειγμάτων το πρόγραμμα του ζητάει το email του χρήστη



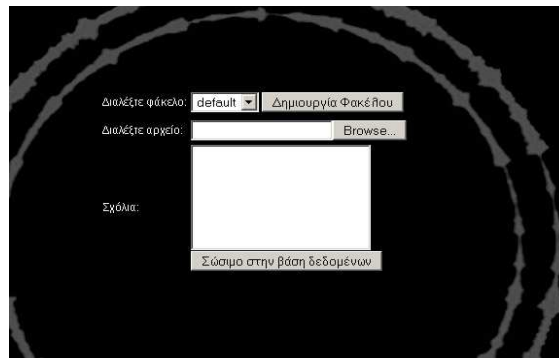
προκειμένου να στείλει ένα mail με χρήσιμες πληροφορίες σε σχέση με την κατασκευή αλλά και την πλοήγηση στην πύλη.

Στην συνέχεια αφού δοθεί και το email από τον χρήστη, ο χρήστης πλοηγείται στην σελίδα ανταλλαγής δειγμάτων. Εκεί του δίνονται δύο επιλογές είτε να φορτώσει ένα δείγμα στην βάση δεδομένων είτε να αναζητήσει ηχητικά δείγματα (εικ. 3.5).



**Εικόνα 3.5: Επιλογή λειτουργίας**

Αν επιλέξει «Φόρτωση ηχητικών δειγμάτων» τότε θα προκύψει μια φόρμα όπου ο χρήστης πρέπει να επιλέξει εάν θα φορτώσει το δείγμα στον default φάκελο που φτιάχνει το πρόγραμμα για αυτόν ή αν θα φτιάξει καινούργιο φάκελο με δική του ονομασία. Αφού κάνει αυτό τότε αναζητά το δείγμα προς φόρτωση και αφήνει και σχόλια σε σχέση με αυτό (εικ. 3.6).



**Εικόνα 3.6 : Φόρτωση δείγματος**

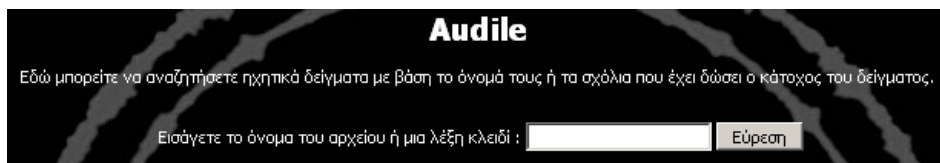
Αφού φορτωθεί το δείγμα ο χρήστης μπορεί να φορτώσει και άλλο πατώντας στον αντίστοιχο σύνδεσμο. Αν επιλέξει «Αναζήτηση ηχητικού δείγματος» τότε θα εμφανιστεί μια λίστα με όλους τους φακέλους των σπουδαστών που έχουν ηχητικά δείγματα. Πατώντας στην επισκόπηση δειγμάτων θα εμφανιστούν όλοι οι φάκελοι με τα ηχητικά δείγματα του συγκεκριμένου φοιτητή και όταν ξαναπατήσει ο χρήστης επισκόπηση δειγμάτων τότε θα του εμφανιστούν όλα τα δείγματα από το συγκεκριμένο φάκελο. Στην συνέχεια πατώντας πάνω στον σύνδεσμο με την κυματομορφή ο χρήστης μπορεί να δει αναλυτικές πληροφορίες σε σχέση με το δείγμα αλλά και την κυματομορφή που προκύπτει από την ανάλυση αυτού του δείγματος (εικ. 3.7). Για να ακούσει ο χρήστης ένα δείγμα πρέπει να πατήσει πάνω στον σύνδεσμο με το αντίστοιχο εικονίδιο της αναπαραγωγής ενώ για να το κατεβάσει πρέπει να πατήσει δεξιά click και στην συνέχεια την επιλογή της αποθήκευσης ανάλογα με τον φυλλομετρητή που χρησιμοποιεί.



**Εικόνα 1.7 : Πληροφορίες**

Για να φορτώσει μια μίξη ο χρήστης ακολουθεί ακριβώς την ίδια διαδικασία αλλά πατώντας αρχικά στον σύνδεσμο «Ανταλλαγή μίξης».

Όταν ο χρήστης θέλει να αναζητήσει ένα δείγμα χωρίς να ξέρει σε ποιον σπουδαστή ανήκει μπορεί να πατήσει στον σύνδεσμο «Αναζήτηση Αρχείου» και τότε θα εμφανιστεί μια μπάρα αναζήτησης προκειμένου να εισάγει μια λέξη κλειδί (εικ. 3.8). Σε αυτή την περίπτωση το Audile θα αναζητήσει δείγματα στα οποία εμπεριέχεται είτε σαν σχόλιο είτε σαν όνομα η λέξη κλειδί που εισήγαγε ο χρήστης και θα του εμφανίσει τα αντίστοιχα δείγματα σε μια λίστα.



**Εικόνα 3.8 : Εύρεση δειγμάτων**

Ένας άλλος σύνδεσμος που υπάρχει είναι αυτός του ραδιοφώνου. Όταν πατηθεί ο σύνδεσμος «Ραδιόφωνο» ο χρήστης πρέπει είτε να πατήσει πάνω στον επόμενο σύνδεσμο «Radio» (εικ 3.9) και στην επόμενη σελίδα τον σύνδεσμο «Listen» είτε να κάνει επικόλληση την διεύθυνση <http://audile.ath.cx:8000/> στο player του.



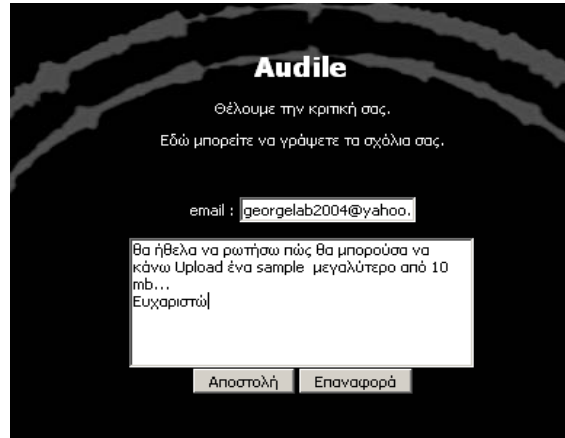
**Εικόνα 3.9 : Ραδιόφωνο**

Το Audile έχει επίσης μια εκτενή λίστα με πολύ χρήσιμους συνδέσμους που παρέχουν πολύ χρήσιμες πληροφορίες σε σχέση με τον ήχο και όχι μόνο. Για να εμφανιστεί αυτή η λίστα αρκεί να πατηθεί ο σύνδεσμος «Σύνδεσμοι» (εικ. 3.10).



**Εικόνα 3.10 : Σύνδεσμοι**

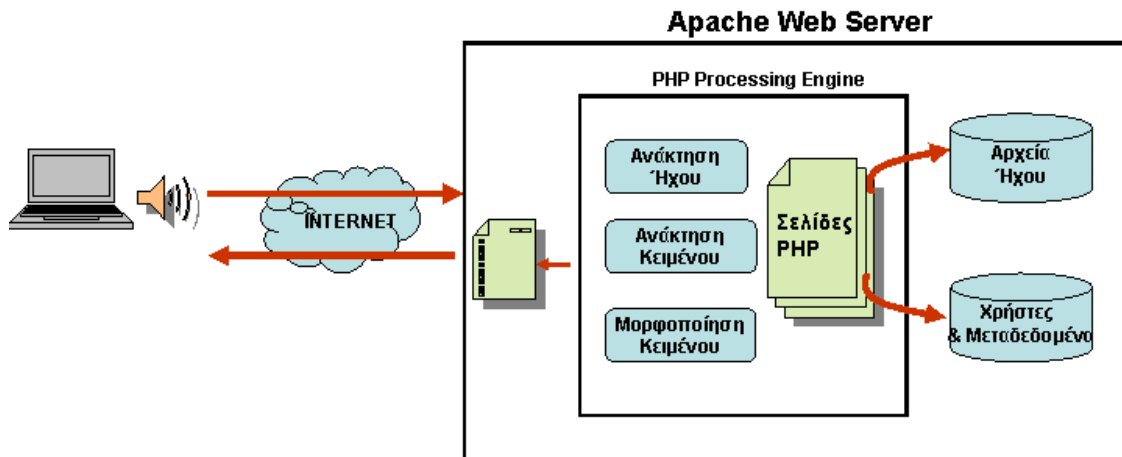
Τέλος στον σχεδιασμό διεπαφής έχει προστεθεί και ένας τελευταίος σύνδεσμος προκειμένου να μπορεί ο χρήστης να επικοινωνήσει με το κατασκευαστή του προγράμματος για πιθανές απορίες ή αναβαθμίσεις. Αυτό γίνεται πατώντας στο σύνδεσμο «Επικοινωνία» όπου στην συνέχεια εμφανίζεται η αντίστοιχη φόρμα επικοινωνίας (εικ. 3.11).



**Εικόνα 3.11 : Επικοινωνία**

## 4 Υλοποίηση

### 4.1 Αρχιτεκτονική στην οποία βασίζεται ο δικτυακός τόπος audile



Εικόνα 4.1: Γενική αρχιτεκτονική του συστήματος

Στο παραπάνω διάγραμμα απεικονίζεται η αρχιτεκτονική στην οποία βασίζεται το Audile. Ειδικότερα, οι σελίδες του Audile παρέχονται από τον εξυπηρετητή Apache Web Server (βλ. 4.2.4). Οι σελίδες αυτές είναι γραμμένες σε γλώσσα PHP. Καθώς ο χρήστης ζητάει μία συγκεκριμένη σελίδα, από το server, ο τελευταίος επεξεργάζεται τις εντολές PHP που περιέχονται στη σελίδα προκειμένου η επιστρεφόμενη στον Internet Browser του χρήστη σελίδα να περιλαμβάνει απλές οδηγίες HTML. Οι PHP εντολές χρησιμεύουν στην είσοδο του χρήστη (login), καθώς και στην πρόσβαση στη βάση δεδομένων.

### 4.2 Εργαλεία Ανάπτυξης

Προκειμένου να υλοποιηθεί το Audile χρησιμοποιήθηκαν τα εξής εργαλεία ανάπτυξης που θα τα εξηγήσουμε παρακάτω:

- **HTML** (Hypertext Markable Language) – Γλώσσα σήμανσης ιστοσελίδων
- **CSS (Cascading Style Sheets)** – Γλώσσα μορφοποίησης ιστοσελίδων
- **JAVASCRIPT** – Τύπος προγραμματισμού που προσθέτει διαδραστικές λειτουργίες σε μια ιστοσελίδα, οι οποίες εκτελούνται στην πλευρά του χρήστη (client-side scripting).
- **PHP** (Hypertext Pre Processor) – Τύπος προγραμματισμού που προσθέτει διαδραστικές λειτουργίες σε μια ιστοσελίδα, οι οποίες εκτελούνται στην πλευρά του εξυπηρετητή (server-side scripting).
- **MySQL** (My Structured Query Language) – Σύστημα διαχείρισης βάσης δεδομένων βασισμένο σε SQL γλώσσα

- **Apache HTTP Server** – Open source πρόγραμμα που επιτρέπει την χρήση του υπολογιστή σαν εξυπηρετητή

Το συγκεκριμένο πρόγραμμα για να είναι όσο το δυνατόν περισσότερο λειτουργικό, αυτό που χρειάζεται είναι ένας αφιερωμένος ηλεκτρονικός υπολογιστής. Ένας server λοιπόν που θα είναι αφιερωμένος καθαρά στο Audile και στην βάση δεδομένων του, θα επιτρέπει στους χρήστες να φορτώνουν (upload) τα αρχεία τους όπως επίσης και να τα καταφορτώνουν (download) αρχεία ήχου από αυτόν.

Στις ενότητες που ακολουθούν γίνεται μία σύντομη παρουσίαση των εργαλείων που χρησιμοποιήθηκαν προκειμένου να κατανοηθεί ο τρόπος υλοποίησης του Audile.

#### 4.2.1 HTML

Στους υπολογιστές η HyperText Markup Language (**HTML**) είναι μια markup γλώσσα όπου markup σημαίνει μια γλώσσα που μπορεί να συνδυάσει κείμενο μαζί με κάποιες άλλες πληροφορίες, οι οποίες ως επί το πλείστον αφορούν τη μορφοποίηση (formatting) του κειμένου. Δημιουργήθηκε προκειμένου να σχεδιαστούν ιστοσελίδες με hypertext για το internet. Όταν βλέπουμε κάποια σελίδα στο διαδίκτυο, συνήθως επικεντρώνουμε την προσοχή μας στο λεκτικό ή οπτικό της περιεχόμενο. Σπάνια αναρωτιόμαστε πώς γίνεται και εμφανίζεται το υλικό αυτό στην οθόνη μας. Η πιο διαδεδομένη άποψη είναι πως το internet είναι μια τεράστια βιβλιοθήκη και η εικόνα που έχουν οι περισσότεροι ταυτίζεται με αυτήν του πραγματικού βιβλίου. Υποθέτουμε ίσως πως η σελίδα που βλέπουμε έρχεται στον υπολογιστή μας πάνω-κάτω όπως και η τηλεοπτική εικόνα. Στην πράξη, υπεύθυνος για την εμφάνισή της είναι ο browser μας (φυλλομετρητής: internet explorer, netscape, opera κλπ). Όταν πληκτρολογούμε σε αυτόν κάποιο url (διαδικτυακή διεύθυνση) ή κάνουμε κλικ σε κάποιο Link (υπερσύνδεσμο) τότε συμβαίνουν (περιληπτικά) τα εξής:

- Η ονομαστική διεύθυνση που δώσαμε μεταφέρεται ως αίτημα στους servers (εξυπηρετητές) της εταιρείας που μας προσφέρει σύνδεση στο Internet και αυτοί εντοπίζουν σε δικές τους ή άλλες βάσεις δεδομένων την αριθμητική διεύθυνση (IP) που αντιστοιχεί στο όνομα αυτό.
- Ο υπολογιστής μας διασυνδέεται με αυτόν τον server και αρχίζουν να μεταφέρονται προς εμάς τα αναγκαία δεδομένα.
- Τα δεδομένα αυτά περιλαμβάνουν: Την περιγραφή δομής της σελίδας, τα περιεχόμενα κείμενα και τις τυχόν εικόνες.
- Ο browser μας συγκεντρώνει τα πιο πάνω στοιχεία και στήνει τη σελίδα στην οθόνη μας.bb

Η πρώτη και πιο διαδεδομένη γλώσσα περιγραφής της δομής μιας ιστοσελίδας είναι η HTML. Η γλώσσα αυτή διαφέρει από τις προγραμματιστικές μια που απλά περιγράφει πού βρίσκεται τι και τίποτε παραπάνω. Νεώτερες γλώσσες είτε αυτόνομες είτε εξαρτώμενες από την HTML (όπως η java, η XML, η PHP ή το Asp) προχωρούν παραπέρα και επιτρέπουν πιο σύνθετη δομή, διαχείριση βάσεων δεδομένων και πολλά άλλα.

Σε αυτές τις σελίδες θα προσπαθήσουμε να μνηθούμε στον τρόπο δημιουργίας ιστοσελίδων με τη χρήση κώδικα HTML. Υπάρχουν βέβαια πολλά προγράμματα που

δημιουργούν σελίδες με απλό και παραστατικό τρόπο. Παρ' όλ' αυτά, η εμπάθυνση στον κώδικα επιτρέπει τη μεγαλύτερη κατανόηση του τι συμβαίνει κι επιτρέπει τη βελτίωση των δημιουργιών μας.

## **4.2.2 PHP**

### **4.2.2.1 Ιστορικά**

Η PHP αρχικά σχεδιάστηκε σαν ένα μικρό σύνολο Perl προγραμματισμού ακολουθούμενο από ένα CGI γραμμένο σε γλώσσα C. Το CGI (Common Gateway Interface) είναι ένα πρωτόκολλο επικοινωνίας που επιτρέπει την επικοινωνία ενός web server με εξωτερικά λογισμικά. Αυτό επιτρέπει στον server να δίνει εντολές από τον browser του client στην εκάστοτε online εφαρμογή.

Ο τρόπος με τον οποίο λειτουργεί το CGI από την σκοπιά του web server είναι ότι κάποιες διευθύνσεις είναι καθορισμένες να εξυπηρετούνται από κάποιο CGI πρόγραμμα. Οποτεδήποτε λαμβάνεται μια αίτηση προς ένα συγκεκριμένο URL το αντίστοιχο πρόγραμμα καλείται με τα δεδομένα που μπήκαν στον client σαν input. Το output από τον browser το επεξεργάζεται ο server και το στέλνει πίσω στον browser.

Επειδή αυτή η τεχνολογία απαιτεί γενικά ένα καθαρό αντίγραφο του προγράμματος για να εκτελεσθεί για κάθε αίτημα της CGI, ο φόρτος εργασίας θα μπορούσε γρήγορα να συντρίψει τους servers του δικτύου.

Αποδοτικότερες τεχνολογίες όπως το mod PERL ή ASP επιτρέπουν στους διερμηνείς των προγραμμάτων να ενσωματωθούν άμεσα στους servers ως ενότητες αποφεύγοντας κατά συνέπεια την συνεχή φόρτωση και επανεκκίνηση των διερμηνέων. Εντούτοις, αυτό ισχύει μόνο για τις αποκαλούμενες γλώσσες υψηλού επιπέδου.

Τέτοιες υπερφορτώσεις μπορούν να αποφευχθούν με τη χρησιμοποίηση των γλωσσών όπως η C. Με τη χρησιμοποίηση της C ή των παρόμοιων συνταγμένων γλωσσών είναι δυνατό να επιτευχθούν τα επίπεδα υψηλότερης αποδοτικότητας, επειδή τέτοια προγράμματα ολοκληρώνουν τον κύκλο εκτέλεσής τους γρηγορότερα από τις ερμηνευμένες γλώσσες με τη λιγότερη υπερφόρτωση λειτουργικών συστημάτων.

### **4.2.2.2 Λειτουργία**

Η PHP τρέχει σε ένα web server. Ο server αυτός έχει σαν είσοδο (input) την PHP η οποία δημιουργεί ιστοσελίδες στην έξοδο (output).

Όταν τρέχει σαν server-side η PHP μπορεί να θεωρηθεί μια εναλλακτική λύση απέναντι στα άλλα συστήματα (ASP.NET της Microsoft ή το Coldfusion κ.α). Για να ανταγωνιστεί η Zend (παραγωγός εταιρεία της PHP) τις άλλες εταιρείες παρήγαγε ένα σύνολο από ολοκληρωμένα κομμάτια κώδικα PHP και καλύτερες πρακτικές. Παρόμοια μοντέλα βασισμένα στον ίδιο τρόπο σκέψης περιέχουν και η CakePHP και η Symphony.

Η αρχιτεκτονική LAMP έγινε πολύ γνωστή στην βιομηχανία του Internet σαν ένας τρόπος ανάπτυξης οικονομικός, αξιόπιστος, κλιμακωτός και ασφαλής ώστε να αναπτύσσονται με αυτόν web εφαρμογές. Η PHP μπορεί να χρησιμοποιηθεί με κάθε

βάση δεδομένων, τρέχει σε όλους τους γνωστούς servers και είναι διαθέσιμη για κάθε τύπο λειτουργικών συστημάτων. Αυτή η λειτουργικότητά της δίνει ένας τεράστιο αριθμό εγκαταστάσεων μέσα στο internet, πάνω από 18 εκατομμύρια internet τοποθεσίες στεγάζονται και εξυπηρετούνται από servers στους οποίους έχει εγκατασταθεί η PHP.

Η PHP προσφέρεται και σε μορφή CLI (command line interface). Το CLI είναι μια μέθοδος επικοινωνίας προς έναν υπολογιστή διάμεσο ενός τερματικού εντολών (text terminal). Επίσης παρέχονται και οι συνδέσεις στις βιβλιοθήκη GUI όπως GTK+ αλλά και στις βιβλιοθήκες text mode προκειμένου να διευκολυνθεί η ανάπτυξη μιας ευρύτερης σειράς του λογισμικού. Χρησιμοποιείται όλο και περισσότερο στο command line για στόχους που φτιάχνονταν παραδοσιακά με Perl ή με shell scripting που είναι μια μέθοδος προγραμματισμού όπου το κάθε shell έχει την γλώσσα του. Κάτι σαν ένα κώδικα html που το γράφουμε και τρέχει αυτομάτως το shell όπως ακριβώς κάνει ο browser με την html.

#### 4.2.2.3 Τι είναι η PHP

Η PHP, της οποίας τα αρχικά αντιπροσωπεύουν το "PHP: Hypertext Preprocessor" είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη εφαρμογών για το Web και μπορεί να ενσωματωθεί στην HTML.

Ένα παράδειγμα:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <?PHP
      echo"Hi,I'm a PHP script!";
    ?>
  </body>
</html>
```

Παρατηρήστε πως αυτό είναι διαφορετικό από ένα script γραμμένο σε άλλες γλώσσες προγραμματισμού όπως η Perl ή η C : Αντί να γράφετε ένα πρόγραμμα με πολλές εντολές για να εξάγετε HTML, γράφετε ένα HTML script με κάποιο ενσωματωμένο κώδικα για να κάνει κάτι (σε αυτή την περίπτωση, να εμφανίζει κάποιο κείμενο). Ο κώδικας PHP είναι εσωκλειστος σε ειδικά tags (ετικέτες) αρχής και τέλους που σας επιτρέπουν να μεταφέρεστε μέσα και έξω από το "PHP mode" (PHP τρόπο λειτουργίας).

Αυτό που διαχωρίζει την PHP από κάτι σαν client-side Javascript είναι ότι ο κώδικας εκτελείται στον server (εξυπηρετητή). Αν είχατε ένα script σαν το παραπάνω στον server σας, ο client θα έπαιρνε τα αποτελέσματα της εκτέλεσης αυτού του script, χωρίς να υπάρχει κανένας τρόπος να καταλάβει τι κώδικας υπάρχει από κάτω. Μπορείτε ακόμη να ρυθμίσετε τον web server σας να χειρίζεται όλα τα HTML αρχεία σας με την PHP, και τότε πραγματικά δεν υπάρχει τρόπος ο χρήστης να καταλάβει τι έχετε κάτω από το μανίκι σας.

Τα καλύτερο πράγμα στην PHP είναι ότι είναι εξαιρετικά απλή για ένα νεοφερμένο αλλά προσφέρει πολλά προηγμένα χαρακτηριστικά για ένα επαγγελματία προγραμματιστή.

Αν και η ανάπτυξη της PHP εστιάζεται σε server-side scripting, μπορείτε να κάνετε πολλά περισσότερα με αυτή.

#### **4.2.2.4 Τι μπορεί να κάνει η PHP;**

Η PHP επικεντρώνεται κυρίως στο server-side scripting, έτσι μπορείτε να κάνετε οτιδήποτε κάνει ένα CGI πρόγραμμα, όπως να μαζέψει δεδομένα, να παράγει δυναμικό περιεχόμενο σελίδων, ή να στείλει και να πάρει cookies. Αλλά η PHP μπορεί να κάνει πολύ περισσότερα.

Υπάρχουν τρεις κύριοι τομείς που χρησιμοποιείται ένα PHP script.

- Server-side scripting. Αυτό είναι το πιο παραδοσιακό και το κύριο πεδίο για την PHP. Χρειάζεστε τρία πράγματα για να δουλέψει αυτό. Τον PHP μεταγλωττιστή (parser) (CGI ή server module), ένα web server (εξυπηρετητή σελίδων) και ένα web browser ("φυλλομετρητή"). Πρέπει να τρέξετε τον web server, με μια συνδεδεμένη εγκατάσταση της PHP. Μπορείτε να προσπελάσετε τα αποτελέσματα του PHP προγράμματος με ένα web browser, βλέποντας την σελίδα PHP μέσα από τον server.
- Command line scripting. Μπορείτε να φτιάξετε ένα PHP script για να το τρέχετε χωρίς server ή browser. Χρειάζεστε μόνο τον PHP μεταγλωττιστή για να την χρησιμοποιήσετε με αυτό τον τρόπο. Αυτός ο τύπος είναι ιδανικός για script που εκτελούνται συχνά με τη χρήση της cron (σε \*nix ή Linux) ή με τον Task Scheduler (στα Windows). Αυτά τα script μπορούν επίσης να χρησιμοποιηθούν για απλές εργασίες επεξεργασίας κειμένου. Δείτε την ενότητα σχετικά με την Command line χρήση της PHP για περισσότερες πληροφορίες.
- Εγγραφή client-side GUI εφαρμογών (Γραφικά περιβάλλοντα χρηστών). Η PHP ίσως να μην είναι η πιο καλή γλώσσα για να γράψει κανείς παραθυρικές εφαρμογές, αλλά αν ξέρετε PHP πολύ καλά και θέλετε να χρησιμοποιήσετε κάποια προχωρημένα χαρακτηριστικά της PHP στις client-side εφαρμογές σας, μπορείτε επίσης να χρησιμοποιήσετε το PHP-GTK για αυτού του είδους τα προγράμματα. Έχετε επίσης τη δυνατότητα να γράφετε cross-platform εφαρμογές με αυτό τον τρόπο. Το PHP-GTK είναι μια επέκταση της PHP και δεν συμπεριλαμβάνεται στην κύρια διανομή. Αν ενδιαφέρεστε για το PHP-GTK, επισκεφτείτε την δική του ιστοσελίδα.

Η PHP μπορεί να χρησιμοποιηθεί σε όλα τα κύρια λειτουργικά συστήματα, συμπεριλαμβανομένου του Linux, πολλών εκδοχών του Unix (HP-UX, Solaris και OpenBSD), Microsoft Windows, Mac OS X, RISC OS και πιθανώς σε άλλα. Η PHP υποστηρίζει επίσης τους Apache, Microsoft Internet Information Server, Personal Web Server, Netscape και iPlanet servers, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd, και πολλούς άλλους webserver. Για την πλειοψηφία των server η PHP έχει ένα module, για τους υπόλοιπους η PHP μπορεί να λειτουργήσει ως ένας CGI επεξεργαστής.

Έτσι με την PHP έχετε την ελευθερία επιλογής ενός λειτουργικού συστήματος και ενός web server. Επιπλέον, έχετε επίσης την ελευθερία να χρησιμοποιήσετε συναρτησιακό



(procedural) ή αντικειμενοστραφή (object oriented) προγραμματισμό ή μια ανάμειξη τους. Αν και η παρούσα έκδοση δεν υποστηρίζει όλα τα πρότυπα χαρακτηριστικά, μεγάλες βιβλιοθήκες κώδικα και μεγάλες εφαρμογές (συμπεριλαμβανομένης και της βιβλιοθήκης PEAR) είναι γραμμένες μόνο με αντικειμενοστραφή κώδικα.

Με την PHP δεν είστε περιορισμένοι να εξαγάγετε HTML. Οι δυνατότητες της PHP συμπεριλαμβάνουν την εξαγωγή εικόνων, αρχείων PDF, ακόμη και animation Flash (χρησιμοποιώντας τα libswf και Ming) παράγονται αμέσως. Μπορείτε επίσης να εξαγάγετε εύκολα οποιοδήποτε κείμενο όπως XHTML και οποιοδήποτε άλλο XML αρχείο. Η PHP μπορεί να δημιουργεί αυτόματα αυτά τα αρχεία και να τα αποθηκεύει στο σύστημα αρχείων, αντί να τα εκτυπώνει, αποτελώντας έτσι μια server-side cache για το δυναμικό σας περιεχόμενο.

Ένα από τα πιο δυνατά και σημαντικά χαρακτηριστικά της PHP είναι η υποστήριξη που έχει για ένα μεγάλο σύνολο βάσεων δεδομένων όπως φαίνεται και στον πίνακα 4.2. Η συγγραφή μιας σελίδας που υποστηρίζει βάσεις δεδομένων είναι εξαιρετικά απλή. Οι εξής βάσεις δεδομένων υποστηρίζονται μέχρι στιγμής:

Adabas D	Ingres	Oracle (OCI7 and OCI8)
dBase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
FilePro (read-only)	mSQL	Solid
Hyperwave	Direct MSSQL	Sybase
IBM DB2	MySQL	Velocis
Informix	ODBC	Unix dbm

#### **Πίνακας 4.2: Βάσεις δεδομένων οι οποίες υποστηρίζονται από τη γλώσσα PHP**

Υπάρχει επίσης μια αφαιρετική επέκταση DBX βάσεων δεδομένων (DBX database abstraction extension) που επιτρέπει να χρησιμοποιηθεί οποιαδήποτε βάση δεδομένων υποστηρίζεται από αυτή την επέκταση. Επιπλέον η PHP υποστηρίζει το ODBC, το Open Database Connection standard (Ανοιχτό πρότυπο Σύνδεσης Βάσεων δεδομένων) έτσι μπορείτε να συνδεθείτε σε οποιαδήποτε βάση δεδομένων που υποστηρίζει αυτό το παγκόσμιο πρότυπο.

Η PHP έχει επίσης υποστήριξη για επικοινωνία με άλλες υπηρεσίες χρησιμοποιώντας πρωτόκολλα όπως LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (στα Windows) και αμέτρητα άλλα. Μπορείτε επίσης να ανοίξετε raw network sockets και να

αλληλεπιδράσετε με οποιοδήποτε άλλο πρωτόκολλο. Η PHP έχει ακόμη υποστήριξη για την περίπλοκη ανταλλαγή δεδομένων WDDX μεταξύ σχεδόν όλων των Web programming γλωσσών. Μιλώντας για δια-επικοινωνία, η PHP υποστηρίζει instantiation αντικειμένων Java και τα χρησιμοποιεί σαν αντικείμενα PHP. Μπορείτε επίσης να χρησιμοποιήσετε την COBRA επέκταση για να προσπελάσετε remote (απομακρυσμένα) αντικείμενα.

### 4.2.3 Ηλεκτρονικές βάσεις δεδομένων

Οι ηλεκτρονικές βάσεις δεδομένων είναι οργανωμένες κατά τέτοιο τρόπο, ώστε να μπορεί κανείς εύκολα και γρήγορα να εισάγει νέα στοιχεία, να παρακολουθεί το σύνολο των πληροφοριών που περιέχουν και να επεξεργάζεται τις πληροφορίες αυτές με ποικίλους τρόπους. Η οργάνωση και γενικά η διαχείριση ηλεκτρονικών βάσεων δεδομένων γίνεται από μια ομάδα εφαρμογών λογισμικού, που ονομάζουμε σύστημα διαχείρισης βάσης δεδομένων (database management system, DBMS). Οι λειτουργίες ενός τέτοιου συστήματος συνοψίζονται στα εξής:

- Ορισμός των δεδομένων και η οργάνωση τους στο μέσο αποθήκευσης (σκληρός δίσκος).
- Διαχείριση των δεδομένων (αυτόματη δημιουργία κλειδιού κτλ.)
- Έλεγχος των δεδομένων
- Τροφοδότηση άλλων εφαρμογών με δεδομένα στην μορφή που αυτές τα ζητούν.

Για την υλοποίηση ενός σχεσιακού μοντέλου με την χρήση DBMS, έχουν δημιουργηθεί εφαρμογές γνωστές ως συστήματα διαχείρισης σχεσιακής βάσης δεδομένων (relational database management systems, RDBMS).

Τα πλεονεκτήματα από την χρήση τέτοιων DBMS συνοψίζονται στα εξής:

- Επεξεργασία των δεδομένων με ολοκληρωμένο και ενιαίο τρόπο
- Προσπέλαση στα δεδομένα από πολλούς διαφορετικούς χρήστες
- Εύκολη μεταφορά των δεδομένων σε άλλες εφαρμογές
- Εύκολη αναπροσαρμογή της οργάνωσης των δεδομένων
- Αποφυγή επανάληψης των δεδομένων
- Ευελιξία στην διαχείριση των πληροφοριών
- Ευκολία στον έλεγχο και στην ασφάλεια των δεδομένων, δηλαδή στις δυνατότητες προστασίας τους από "αδιάκριτα μάτια".

Τα σύγχρονα DBMS έχουν στο "οπλοστάσιό" τους εργαλεία , που μας επιτρέπουν:

- να δημιουργούμε πίνακες,

- να εισάγουμε και να μεταβάλλουμε δεδομένα στους πίνακες εύκολα, μέσα από οθόνες φορμών,
- να αντλούμε πληροφορίες με την δημιουργία ερωτημάτων,
- να εκτυπώνουμε τις πληροφορίες που θέλουμε χρησιμοποιώντας αναφορές.

Ας δούμε αναλυτικά αυτά τα εργαλεία:

**Όνομα:** Κάθε πεδίο φέρει ένα αντιπροσωπευτικό όνομα, το οποίο είναι μοναδικό μέσα στον πίνακα

**Τύπος:** Ο τύπος κάθε πεδίου χαρακτηρίζεται το είδος των δεδομένων που θα καταχωρηθούν σε αυτό. Τα πεδία μπορούν να είναι αριθμητικά, ημερομηνίας, κειμένου και λογικά. Τα λογικά πεδία αναφέρονται στο αν ισχύει ή όχι μια συνθήκη.

**Εύρος:** Το εύρος ενός πεδίου καθορίζει το μέγιστο αριθμό χαρακτήρων ή ψηφίων που μπορεί να περιέχει το πεδίο για κάθε εγγραφή.

**Πεδίο κλειδί:** Πρέπει να καθορίσουμε ποιο θα είναι το πεδίο (ή τα πεδία) που θα παίζουν το ρόλο του κλειδιού. Θυμόμαστε πάντα ότι το κλειδί πρέπει να φέρει μοναδική τιμή για κάθε εγγραφή.

Οι **φόρμες (forms)** είναι το εργαλείο που χρησιμεύει κυρίως για την εισαγωγή (entry) και την προβολή δεδομένων. Μπορούμε να τις χρησιμοποιήσουμε όμως και για την τροποποίησή τους.

Τα ερωτήματα (queries) είναι τα εργαλεία απαραίτητα για την αναζήτηση πληροφοριών μέσα στην βάση δεδομένων, από έναν ή περισσότερους πίνακες. Το αποτέλεσμα από την εκτέλεση ερωτημάτων έχει την μορφή και την δομή ενός πίνακα. Ωστόσο, ένα ερώτημα μπορεί να εμφανίζει τα αποτελέσματα του και μέσα από μια φόρμα.

Τα ερωτήματα λειτουργούν με τον καθορισμό συνθηκών και κριτηρίων που θέλουμε να ισχύουν για τα προς εμφάνιση δεδομένα. Κάθε ερώτημα δημιουργείται δυναμικά κατά την στιγμή της εκτέλεσής του. Αυτό σημαίνει ότι για ένα ερώτημα δεν αποθηκεύονται τα δεδομένα των αποτελεσμάτων του στην βάση, αλλά μόνο τα κριτήρια που περιλαμβάνει.

#### 4.2.3.1 MySQL

Η **MySQL** είναι ένα πολύπλοκο, πολλών χρηστών (multi-user), σύστημα διαχείρισης βάσεων δεδομένων SQL (DBMS-Database management system) με περισσότερες από έξι εκατομμύρια εγκαταστάσεις. Η MySQL αβ καθιστά την MySQL διαθέσιμο δωρεάν με άδεια GNU General Public License (GPL), αλλά υπάρχει επίσης μια δεύτερη άδεια στο πλαίσιο των παραδοσιακών ιδιοκτητών συμφωνιών χορήγησης αδειών για τις περιπτώσεις όπου η προοριζόμενη χρήση είναι ασυμβίβαστη με το GPL.

Αντίθετα από τα προγράμματα όπως ο Apache, όπου το λογισμικό αναπτύσσεται από μια δημόσια κοινότητα, και τα πνευματικά δικαιώματα στο codebase είναι κύριος από τους μεμονωμένους συντάκτες του, η MySQL υποστηρίζεται από μια ενιαία εταιρεία με σκοπό το οικονομικό κέρδος, την σουηδική επιχείρηση MySQL αβ, η οποία κρατά τα πνευματικά δικαιώματα στο μεγαλύτερο μέρος του codebase. Η επιχείρηση

αναπτύσσει και διατηρεί το σύστημα της πουλώντας συμβάσεις υποστήριξης και υπηρεσιών, καθώς επίσης και εξουσιοδοτημένα αντίγραφα MySQL, και απασχολεί χιλιάδες άτομα σε όλο τον κόσμο που συνεργάζονται μέσω του Διαδικτύου. Η MySQL αβ ιδρύθηκε από τους David Axmark, Allan Larsson, και Michael "Monty" Widenius.

#### **4.2.3.2 Επιδόσεις**

Οι κατασκευαστές της MySQL προσπαθώντας να τελειοποιήσουν την λειτουργία της βάσης της δώσανε κάποια χαρακτηριστικά που θα αναφέρουμε εκτενώς παρακάτω.

##### ***Multiple Table Handlers***

Η MySQL δίνει την δυνατότητα στους χρήστες της να χρησιμοποιούν οποιοδήποτε αλγόριθμο από τους τρεις παρακάτω προκειμένου να εκτελούνται οι εντολές. Οι τρεις αυτοί τύποι αλγορίθμων είναι ο HEAP, InnoDB και ο MyISAM. Ο καθένας από τους παρακάτω εμφανίζει προτερήματα αλλά και αδυναμίες έναντι του άλλου και επειδή μια απλή βάση δεδομένων μπορεί να περιλαμβάνει πολλούς πίνακες, ο καθένας με τον ρόλο του, η MySQL επιτρέπει την χρήση διαφορετικών table handlers μέσα στην ίδια βάση δεδομένων.

##### ***Query Caching***

Το query cache της MySQL είναι μια μεγάλη αναβάθμιση σε σχέση με των χρόνους που έπαιρναν τα queries στην βάση δεδομένων. Όταν λοιπόν το έχουμε ενεργοποιημένο η MySQL θα αποθηκεύσει στην μνήμη ότι έχουμε κάνει select

μαζί με τις τιμές που καλέσαμε. Καθώς τα επακόλουθα queries θα εκτελούνται η MySQL θα συγκρίνει τις τιμές με τα προηγούμενα queries και αν ταιριάζουν θα δώσει προτεραιότητα στα στοιχεία με τον πιο μεγάλο αριθμό δεδομένων και θα απορρίψει τα υπόλοιπα διότι θα έχουν τα ίδια στοιχεία αλλά λιγότερα και έτσι θα εκτελέσει αυτό το query που της χρειάζεται σε λιγότερο χρόνο. Για να περιοριστούν φαινόμενα απαρχαιωμένων αποτελεσμάτων αυτομάτως η MySQL αφαιρεί τα κρυμμένα αποτελέσματα που κρατάει στην μνήμη και τα επαναφέρει με την επόμενη κλήση.

##### ***Full-Text Indexing and Searching***

Αυτό το χαρακτηριστικό αυξάνει την λειτουργία εκμαίευσης πληροφοριών από στήλες κειμένου της βάσης. Επίσης το χαρακτηριστικό αυτό μας επιτρέπει να παράγουμε αποτελέσματα με βάση το πώς αυτά παρουσιάζονται και σε σχέση με το κατά πόσο το query ταιριάζει με την συγκεκριμένη σειρά.

##### ***Replication***

Με αυτό το χαρακτηριστικό η MySQL μπορεί να αντιγράψει ακριβώς μια βάση που βρίσκεται σε ένα MySQL server σε μια άλλη. Αυτό είναι πολύ σπουδαίο προνόμιο έναντι άλλων βάσεων και αυτό γιατί άμα π.χ. η master βάση μας παρουσιάσει κάποιο πρόβλημα μπορούμε γρήγορα και εύκολα να την αντικαταστήσουμε με μια βάση ουσιαστικά καθρέφτη της master.

##### ***Platform Flexibility***

Η MySQL την συγκεκριμένη στιγμή έχει την δυνατότητα να τρέξει πάνω σε 12 διαφορετικές πλατφόρμες. Αυτές είναι : Dec OSF, FreeBSD, IBM AIX, HP-UX, LINUX, MAC OS X, Novell Netware, OpenBSD, QNX, SGI Irix, Solaris και Microsoft Windows.

### *Advanced Security and Configuration Options*

Με τις νέες δυνατότητες της MySQL ο χρήστης μπορεί να ελέγχει παραμέτρους όπως:

Την τοποθεσία όπου τα δεδομένα της βάσης θα σώζονται, την κυρίως γλώσσα, την κύρια θύρα κ.α.

- Το ποσοστό της μνήμης που είναι αφιερωμένο είτε σε τυχόν απειλές του διαδικτύου είτε για να εκτελεστεί μια εντολή.
- Σε διάφορους τομείς δικτυακών εφαρμογών όπως το πόσο γρήγορα θα συνδεθεί ή όχι η βάση, την κατανόηση DNS κ.α.

Όσο αφορά την προστασία η MySQL επιτρέπει πλέον στον χρήστη να ορίσει τα παρακάτω:

- Τον μέγιστο αριθμό εντολών, αναβαθμίσεων και συνδέσεων που επιτρέπονται την ώρα.
- Το κατά πόσο ο χρήστης πρέπει να φανερώσει το SSL πιστοποιητικό προκειμένου να συνδεθεί στη βάση.
- Τις δυνατότητες που έχουν δοθεί σε κάθε χρήστη σε σχέση με το τι μπορεί να σβήσει και σε τι βάθος μπορεί να χειριστεί την βάση.

### **4.2.4 APACHE**

Από την στιγμή που αναφέραμε παραπάνω πώς η PHP εκτελείται στον server πρέπει να αναφέρουμε λίγα πράγματα και για τον software server που χρησιμοποιήθηκε που δεν είναι άλλος από τον apache. Εξυπηρετητής ή διακομιστής (αγγλικά: server) είναι ο υπολογιστής που αναλαμβάνει τον ρολό να παρέχει διάφορες υπηρεσίες «εξυπηρετεί» άλλους πελατειακούς υπολογιστές (clients) ενός τοπικού δικτύου ή ακόμα και στο Παγκόσμιο Ιστό. Ο εξυπηρετητής συνήθως διαφέρει ως προς την σύνθεση του από άλλους κοινούς υπολογιστές μιας και οι δυνατότητες του σαφώς είναι αναβαθμισμένες. Κύρια χαρακτηριστικά ενός εξυπηρετητή είναι οι επεξεργαστές που υποστηρίζει και που χρησιμοποιεί για την επεξεργασία των πολλαπλών δεδομένων που δέχεται, οι γρήγοροι και μεγάλης χωρητικότητας σκληροί δίσκοι αλλά και οι ταχύτητες μνήμης που υποστηρίζει. Συνηθίζεται να συνοδεύεται με σύστημα διπλής τροφοδοσίας (dual power supply) και από συσκευή παροχής αδιάλειπτης ενέργειας (UPS) για μεγαλύτερη αξιοπιστία και σιγουριά της παροχής υπηρεσιών του. Ο Apache HTTP Server είναι ένας open-source HTTP server για σύγχρονα λειτουργικά συστήματα όπως UNIX, MS Windows, Macintosh και Netaware.

The Apache Software Foundation (ASF) είναι μια μη κερδοσκοπική εταιρεία (που ταξινομείται ως 501(c)3 στις Η.Π.Α) προκειμένου να υποστηρίξει τις εφαρμογές λογισμικού Apache, συμπεριλαμβανομένου του κεντρικού υπολογιστή HTTP Apache. Το ASF διαμορφώθηκε από την ομάδα Apache και ενσωματώθηκε στο Delaware των ΗΠΑ τον Ιούνιο του 1999.

Το ίδρυμα λογισμικού Apache είναι αποκεντρωμένη κοινότητα προγραμματιστών. Το λογισμικό που παράγεται διανέμεται υπό τους όρους της Apache License και είναι ένα δωρεάν λογισμικό ή αλλιώς open source. Τα προγράμματα Apache χαρακτηρίζονται από μια συνεργατική, βασισμένη στη πλειοψηφούσα γνώμη, διαδικασία. Κάθε πρόγραμμα ρυθμίζεται από μια εθελοντική ομάδα προγραμματιστών. Το ASF λειτουργεί με ένα αξιοκρατικό σύστημα, που υπονοεί ότι η ιδιότητα μέλους στο ίδρυμα χορηγείται μόνο στους εθελοντές που έχουν συμβάλει ενεργά στα προγράμματα Apache.

Η ιστορία του ιδρύματος λογισμικού Apache συνδέεται με τον κεντρικό υπολογιστή HTTP Apache, η εργασία για τον οποίο άρχισε το 1994. Μια ομάδα οκτώ υπεύθυνων για την ανάπτυξη άρχισε στην ενίσχυση του NCSA HTTPd daemon. Ήταν ο Brian Behlendorf, Roy Fielding, rob Hartill, David Robinson, Cliff Skolnick, Randy Terbush, Robert S. Thau και Andrew Wilson με τις πρόσθετες συνεισφορές από Eric Hagberg, το Frank Peters και Nicolas Pioch.

Ο Apache server απελευθερώθηκε τον Απρίλιο του 1995. Το 1999, τα μέλη της ομάδας Apache διαμόρφωσαν το ίδρυμα ώστε να υποστηρίξουν τον server HTTP Apache. Το ASF αποτελείται από το σύνολο 151 μελών και περίπου 1000 συνεργατών από το 2005.

Ο Apache είναι ο πιο διαδεδομένος server στο Internet από το 1996 και αυτό οφείλεται στο ότι είναι πολύ σταθερός και ταυτόχρονα διατίθεται δωρεάν καθώς είναι open-source. Τον Νοέμβριο του 2005 η Netcraft Web Server Survey παρατήρησε πως πάνω από 70% των website στεγάζονται κάτω από των Apache

Μεταξύ των στόχων του ASF είναι να παρασχεθεί η νομική προστασία στους εθελοντές που εργάζονται στα προγράμματα Apache, και να αποτραπεί το εμπορικό σήμα Apache από τη χρησιμοποίηση από άλλες οργανώσεις χωρίς άδεια.

### 4.3 Ανταλλαγή ηχητικού περιεχομένου

Ας δούμε λοιπόν πως χρησιμοποιήθηκαν τα παραπάνω εργαλεία ώστε να πραγματοποιηθεί το audile. Έχουμε: Apache, PHP, Html, Javascript και την MySQL. Παρακάτω θα δούμε ένα απλό PHP script.

```
<?PHP
```

```
function connect_mysql()
```

```
{
```

```
// connect to the db
```

```
$link = mysql_connect("localhost","root","yo") or die ("Could not connect to mysql because  
".mysql_error());
```

```
// select the database
```

```
mysql_select_db("mydb");
```

```
return $link;
```

```
}
```

```
?>
```

Εδώ έχουμε μια απλή εντολή γραμμένη σε PHP γλώσσα προγραμματισμού που ανοίγει και κλείνει με αυτά τα 2 σύμβολα <?PHP και ?>. Στην συνέχεια φτιάχνουμε μια εντολή `function connect_mysql() {}` την οποία εντολή θα την καλέσουμε από ένα άλλο script φτιαγμένο για αυτήν τη δουλειά, να καλεί τις εντολές. Με αυτήν την εντολή τώρα μπορεί ο χρήστης να συνδεθεί με την βάση δεδομένων της MySQL. Συγκεκριμένα λέμε στην βάση να συνδεθούμε μαζί της μέσω της εντολής

```
$link = mysql_connect("localhost","root","yo") or die ("Could not connect to mysql because ".mysql_error());
```

της PHP και στην συνέχεια λέμε στην MySQL συγκεκριμένα πια βάση θέλουμε.

```
// select the database
```

```
mysql_select_db("mydb");
```

Τέλος να εμφανιστεί αυτή στον browser.

```
return $link;
```

Αυτή είναι και η φιλοσοφία που κρύβεται πίσω από το `audile`. Οτιδήποτε θελήσουμε να κάνουμε καλείται με μια εντολή (`function`) που έχουμε δημιουργήσει ειδικά για αυτόν τον λόγο και έπειτα χρησιμοποιούμε την πληθώρα μεταβλητών της ίδιας της PHP. Αυτές είναι που έπονται του `$`. Το να αναλύσουμε και να προσπαθήσουμε να μάθουμε να προγραμματίζουμε σε PHP γλώσσα δεν είναι ο σκοπός μας. Σκοπός μας είναι να εξοικειωθούμε με τον κόσμο του internet αντιλαμβάνοντας από τι αποτελείται:

Τα πάντα λοιπόν τρέχουν μαζί με τον `apache` ο οποίος πραγματοποιεί αυτά που εμείς έχουμε προγραμματίσει να κάνει.

Έτσι εμείς τώρα έχουμε 9 scripts που αλληλοκαλούνται ανάλογα με το τι εμείς σαν χρήστες διαλέγουμε να κάνουμε.

**ss.PHP**

**functions.PHP**

**htmlrep.PHP**

**content.PHP**

## **classAudioFile.PHP**

## **id3.class.PHP**

## **style.css**

## **smtp.PHP**

## **index.html**

Αναλυτικά:

## **ss.PHP**

Σε αυτό το αρχείο περιέχονται όλες οι αναζητήσεις των εντολών του κώδικά μας. Από το ss.PHP ουσιαστικά καλούμε τις εντολές (functions) που περιέχονται μέσα στο functions.PHP. Τα δύο αυτά αρχεία παίζουν το ρόλο της κύριας λογικής του προγράμματος.

## **functions.PHP**

Όλες οι εντολές (functions) του προγράμματος. Οποιαδήποτε και αν είναι η απόφαση μας σε σχέση με την περιήγησή μας όπως και το πώς και το πού αποθηκεύονται τα samples μέσα στη βάση μας ακόμα και το τι θα μας επιστρέφει στον browser είναι προγραμματισμένο και βρίσκεται μέσα στο functions.PHP και καλείται από το ss.PHP.

## **htmlrep.PHP**

Σε αυτό το αρχείο είναι όλος ο html κώδικας του προγράμματος. Το πώς και που θα εμφανίζονται οι εικόνες, τα γράμματα και οι πίνακες ρυθμίζονται από εδώ.

## **content.PHP**

Εδώ εμπεριέχεται ο κώδικας που εμφανίζεται μέσα στο content. Το content είναι αυτό που μας δείχνει ο browser μας εφόσον έχει γίνει το login από τον χρήστη. Τα κείμενα και οι φόρμες λοιπόν που έπονται του login βρίσκονται μέσα στο content.PHP.

## **classAudioFile.PHP**

Αυτό είναι ένα αυτόνομο αρχείο, ένα class όπως λέγεται, το οποίο είναι υπεύθυνο να κατανοήσει και να καταχωρήσει τις τιμές σε σχέση με τα χαρακτηριστικά του ψηφιακού ήχου όπως την δειγματοληψία, kbps, διάρκεια, εύρος bit, path, κανάλια κ.α. Τέλος το classAudioFile.PHP είναι αυτό που αποκωδικοποιεί τη ένταση του ήχου σε σχέση με τον χρόνο και μας παρέχει την εικόνα στα wav αρχεία που ανεβάζουμε. Το συγκεκριμένο αρχείο είναι και το μόνο που επικοινωνεί με το id3class.PHP.

## **id3.class.PHP**

Το id3 είναι ένα είδος καταχωρητή μεταδεδομένων(metadata). Το id3 μας δίνει χρήσιμες πληροφορίες σε ότι αφορά το mp3 format όπως τον τίτλο, καλλιτέχνη, album, αριθμό κομματιού ή και άλλες πληροφορίες που αφορούν το αρχείο το ίδιο



που θα αποθηκευτεί. Έτσι και στο audile το id3.class.PHP κρατά όλα τα metadata και τα παρέχει όποτε του ζητηθούν.

### style.css

Η css είναι μια γλώσσα stylesheet (stylesheet language) που χρησιμοποιείται για να περιγράψει την παρουσίαση ενός εγγράφου που γράφεται σε μια markup language. Η πιο γνωστή εφαρμογή του είναι να περιγράψει (style) ιστοσελίδες φτιαγμένες σε HTML και XHTML, αλλά η css μπορεί να εφαρμοστεί σε οποιοδήποτε XML, SVG, XUL αρχείο. Οι προδιαγραφές διατηρούνται από την World Wide Web Consortium (W3C).

Στο audile λοιπόν το style.css κάνει αυτό που ακριβώς περιγράψαμε παραπάνω. Από εκεί ελέγχουμε το ύψος του πίνακα, των fonts και τα χρώματα επικοινωνώντας με το htmlrep.PHP.

### smtp.PHP

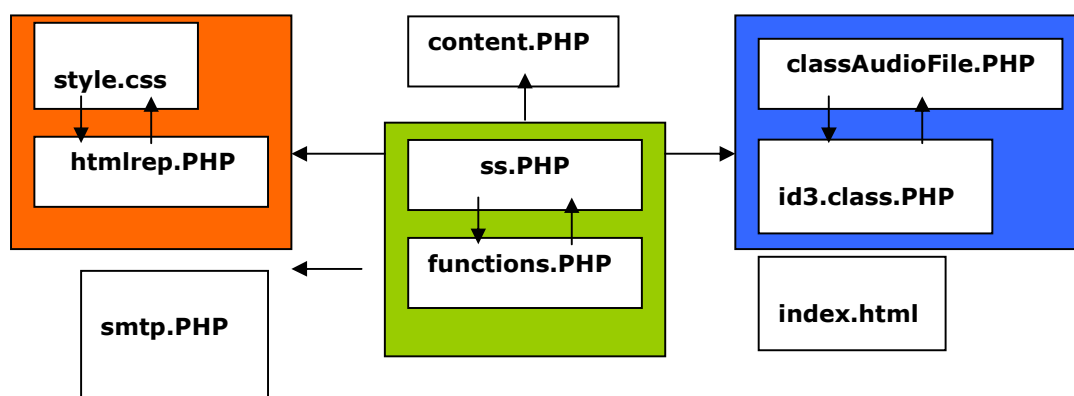
Το smtp είναι ουσιαστικά και αυτό ένα class προκειμένου να στείλει κάποιος e-mail και internet fax μέσω του internet.

Εδώ χρησιμοποιείται ώστε να σταλεί το e-mail που στέλνεται σε κάθε χρήστη με το που κάνει login και γράφει την ηλεκτρονική του διεύθυνση στην αντίστοιχη φόρμα για τα e-mail που εμφανίζεται.

### index.html

Το index.html είναι το αρχείο που περιέχει τον κώδικα από το flash. Το flash είναι ένα λογισμικό πολύ χρήσιμο για την δημιουργία πιο καλλιτεχνικών ιστοσελίδων από την άποψη του design και animation. Το πλεονέκτημα του είναι πως είναι πολύ εύχρηστο και λειτουργικό σε σχέση με τα υπόλοιπα προγράμματα που κάνουν την ίδια δουλειά. Ο κώδικας λοιπόν που χρειάζεται για να τρέχει ένα animation μετά την δημοσίευση (publish) του από το Flash εμπεριέχεται μέσα στο index.html. Εδώ βρίσκεται όλη η εισαγωγή (intro) της ιστοσελίδας. Το συγκεκριμένο αρχείο είναι αυτόνομο δεν καλείται ούτε καλεί κανένα. Απλώς είναι αυτό που εμφανίζεται πρώτο στην ιστοσελίδα.

Παρακάτω στην εικόνα 4.3 δείχνεται το πώς αλληλεπικοινωνούν τα αρχεία προκειμένου να δώσουν το επιθυμητό αποτέλεσμα.



### **Εικόνα 4.3: Επικοινωνία Αρχείων**

#### **4.4 Πλοήγηση στο πρόγραμμα**

Καλό θα ήταν σε αυτό το σημείο να εξηγήσουμε βήμα-βήμα το πώς δουλεύει το πρόγραμμα καθώς εμείς πλοηγούμαστε σε αυτό.

Πρώτη σελίδα που ζητάμε από τον browser μας να βρει είναι αυτή <http://62.38.194.54/final/index.html>. Αυτή η σελίδα όπως προαναφέραμε είναι μια εισαγωγή ώστε να μπούμε στην κυρίως σελίδα που είναι αυτή <http://62.38.194.54/final/ss.PHP>. (Οι αριθμοί που βλέπετε δεν είναι τίποτα άλλο από την ip διεύθυνση του τερματικού πάνω στο οποίο τρέχει αυτή την στιγμή ο Apache. Κάθε διεύθυνση στο internet μεταφράζεται σε κάποιους αριθμούς που δεν είναι άλλοι από την ip διεύθυνση του κάθε τερματικού η οποία μπορεί να είναι είτε στατική είτε δυναμική(DHCP). Η στατική είναι πάντα η ίδια οπότε χρησιμοποιείτε πολύ από τους servers και η δυναμική είναι αυτή που έχουν συνήθως τα απλά τερματικά όταν μπαίνουν στο Internet και αλλάζει κάθε φορά που εισέρχονται σε αυτό). Τώρα σε αυτή την σελίδα έχουμε τα εξής:

#### **Login**

Εδώ πρέπει να κάνουμε login και ώστε να μπορούμε να ανεβάσουμε είτε να ακούσουμε κάποιο sample ή μίξη.

Ο κώδικας που απαιτείται από το `ss.PHP` είναι ο παρακάτω αφού προηγουμένως έχουν οριστεί και τα `functions` που χρησιμοποιούμε στο `functions.PHP`.

## functions.PHP

```
function am_initials_are_valid($user_am,$initials){
$link = connect_mysql();
$query =
"SELECT user_am, initials FROM school_users WHERE user_am =
'".$user_am."' && initials = '".$initials."'";
$result = mysql_query($query, $link) or die(mysql_error());
$num_rows = mysql_num_rows($result);

if ($num_rows == 1){

return TRUE;
}

else
{
if(am_initials_are_valid_gr($user_am,$initials)){
return TRUE;
}
else {
return FALSE;
}
}
}

if(isset($_SESSION['logged-in'])){

$content = '';

$no_logins = get_no_logins($_SESSION['user_id']);
$full_name = get_name($_SESSION['user_id']);

$bar = 'Μέχρι στιγμής έχετε συνδεθεί '.$no_logins.'
φορές σαν '.$full_name.' <br>';
$_SESSION['bar'] = $bar;

if(isset($_POST['set_mail'])){
if(strlen(get_email($_SESSION['user_id'])) < 2){
set_email($_SESSION['user_id'], $_POST['email']);
send_mail($_POST['email']);
$content .= 'Το μήνυμα έχει σταλεί!<br>';
}
}
}
```

## ss.PHP

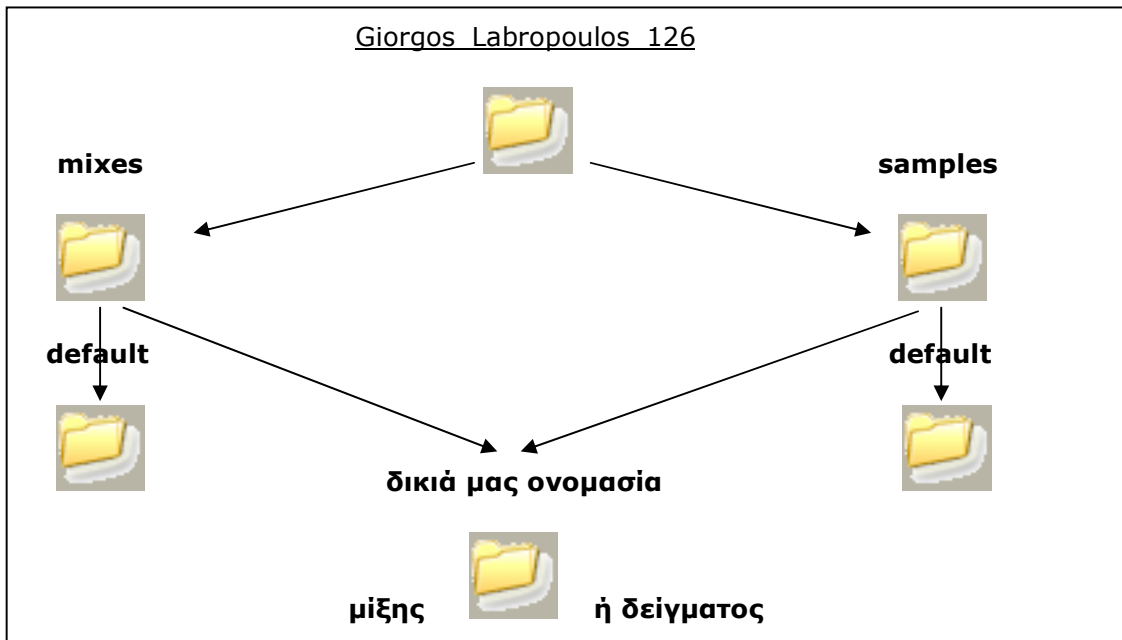
```
if(isset($_POST['login'])) {  
  
    if(am_initials_are_valid($_POST['user_am'], $_POST['initials'])) {  
  
        $_SESSION['logged-in'] = '1';  
        $_SESSION['user_id'] = get_user_id($_POST['user_am']);  
        $no_logins = get_no_logins($_SESSION['user_id']) + 1;  
        set_no_logins($_SESSION['user_id'], $no_logins);  
  
    }  
  
    else {  
  
        $content = 'Παρακαλώ κάντε log-in για να δείτε  
                    τα αρχεία της βάσης δεδομένων';  
        $bar = get_content('login_error');  
        echo get_page($content, $bar);  
        die();  
  
    }  
  
}
```

Έτσι εδώ έχουμε τα δυο αυτά ενδεχόμενα. Αν τα αρχικά και το Α.Μ. που γράφει ο χρήστης είναι σωστά τότε θα δει το πρόγραμμα πόσες φορές έχει κάνει login ο χρήστης και θα σώσει στην βάση τον αριθμό αυτό συν 1 που είναι η τελευταία φορά. Αλλιώς αν είναι λάθος ή τα αρχικά ή το Α.Μ. θα μας βγάλει ένα login error στην ίδια σελίδα που βρισκόμαστε και θα συνεχίσει να μας λέει *'Παρακαλούμε συνδεθείτε για να δείτε τα αρχεία της βάσης.'*

Το επόμενο βήμα είναι να κοιτάξει το πρόγραμμα εάν αυτή είναι η πρώτη φορά που κάνουμε login. Σε αυτή την περίπτωση θα μας βγάλει μια φόρμα όπου θα μας ζητήσει συμπληρώσουμε το email μας και όταν το κάνουμε θα στείλει ένα email με πληροφορίες που σχετίζονται με την πλοήγηση στο Audile και θα μας εμφανίσει στην οθόνη μας 'Το μήνυμα έχει σταλεί!'.

```
function set_email($user_id, $email)  
{  
    $link = connect_mysql();  
    $query = "UPDATE school_users SET email = '". $email. "'  
              WHERE user_id='". $user_id. "'";  
    $result = mysql_query($query, $link) or die (mysql_error());  
}
```

Επίσης θα μας φτιάξει ένα φάκελο και τους υποφακέλους του με το όνομα μας και το Α.Μστο shared directory του server μας . όπως φαίνεται στην εικόνα 4.4.



**Εικόνα 4.4: Αρχαιοθέτηση Φακέλων**

Και ο κώδικας στο functions.PHP.

```
function get_no_logins($user_id){

$name = get_user_name($user_id);
$surname = get_user_surname($user_id);
$am = get_user_am($user_id);

$link = connect_mysql();
$query = "SELECT no_logins FROM school_users
        WHERE user_id='".$user_id."'";
$result = mysql_query($query,$link) or die (mysql_error());
$no_logins = mysql_result($result,0,"no_logins");
//connect mysql get no_logins field

    if ($no_logins >=1)
    {
return $no_logins;
}
else
{
```

```

mkdir("sample_library/".$name.'_'. $surname.'_'. $user_id,
      "0777");
mkdir("sample_library/".$name.'_'. $surname.'_'. $user_id.
      "/samples", "0777");
mkdir("sample_library/".$name.'_'. $surname.'_'. $user_id.
      "/mixes", "0777");
mkdir("sample_library/".$name.'_'. $surname.'_'. $user_id.
      "/samples/default", "0777");
mkdir("sample_library/".$name.'_'. $surname.'_'. $user_id.
      "/mixes/default", "0777");

add_dir($user_id, "default", "samples");
add_dir($user_id, "default", "mixes");
return $no_logins;
}
}

```

### Ανταλλαγή Sample ή Μίξης

Επόμενο βήμα στην πλοήγησή μας είναι να διαλέξουμε ένα από τα 5 κύρια links που υπάρχουν στο htmlrep.PHP. Η λογική πίσω από το Ανταλλαγή Sample και το Ανταλλαγή Μίξης είναι ακριβώς η ίδια με την μόνη διαφορά πως για μίξη το μεγαλύτερο επιτρεπτό μέγεθος προς upload είναι 15MB ενώ για ένα ηχητικό δείγμα είναι 5mb.

Στην σελίδα αυτή μπορούμε είτε να κάνουμε upload κάποιο sample ή να βρούμε ένα sample της αρεσκείας μας. Ας δούμε πρώτα την περίπτωση που κάνουμε upload ένα δείγμα ή μια μίξη.

Όπως είπαμε και προηγουμένως έχουμε με το που κάναμε login τον προσωπικό μας φάκελο με το όνομά μας και μέσα ένα φάκελο που ονομάζεται default. Τώρα εδώ μας δίνετε η επιλογή να δημιουργήσουμε ένα νέο φάκελο με το όνομα που εμείς επιθυμούμε. Πατώντας πάνω στο «Δημιουργία φακέλου» θα μας εμφανιστεί μια κενή μπάρα όπου εκεί να δώσουμε το όνομα του καινούργιου φακέλου. Αμέσως μετά αυτός θα εμφανιστεί στο drop down menu και θα μπορούμε να τον επιλέξουμε. Στην συνέχεια μπορούμε άμα θέλουμε να προσθέσουμε και σχόλια έτσι ώστε να γίνεται το δείγμα πιο πλούσιο σε στοιχεία και με τον τρόπο αυτό πιο προσιτό στους υπόλοιπους χρήστες.

Παρακάτω βλέπουμε τον κώδικα που απαιτείται για τα παραπάνω, μέσα στο function.PHP.

```

function add_dir($user_id, $dir_name, $dir_type)
{
$link = connect_mysql();
$query = "INSERT INTO directories (user_id,dir_name,dir_type)
VALUES ('".$user_id."', '".$dir_name."', '
        ".$dir_type."' )";
$result = mysql_query($query,$link) or die (mysql_error());

$query = "SELECT dir_id FROM directories WHERE dir_type=
        '".$dir_type."' && dir_name = '".$dir_name."'";
$result = mysql_query($query, $link) or die(mysql_error());
$dir_id = mysql_result($result,0,"dir_id");
return $dir_id;
}

function get_dir_name($dir_id, $dir_type){
$link = connect_mysql();

$query = "SELECT dir_name FROM directories WHERE dir_type=
        '".$dir_type."' && dir_id = '".$dir_id."'";
$result = mysql_query($query, $link) or die(mysql_error());
$dir_name = mysql_result($result,0,"dir_name");
return $dir_name;
}

function get_file_name($file_id){
$link = connect_mysql();

$query = "SELECT filename FROM files
        WHERE file_id='".$file_id."'";
$result = mysql_query($query, $link) or die(mysql_error());
$file_name = mysql_result($result,0,"filename");
return $file_name;
}

```

```

}

function get_file_url($file_id){

$link = connect_mysql();

$query = "SELECT url FROM files WHERE file_id='".$file_id.'"";
$result = mysql_query($query, $link) or die(mysql_error());
$url = mysql_result($result,0,"url");
return $url;

}

function file_exist($filename, $dir_id, $file_type){

$link = connect_mysql();

$query = "SELECT file_id FROM files WHERE filename='".$filename.'"
        ' && dir_id='".$dir_id.'" && type='".$file_type.'"";

$result = mysql_query($query, $link) or die(mysql_error());
//echo mysql_result($result,0,"file_id");

// strlen(file_id) <= 0
if($result && strlen(mysql_result($result,0,"file_id")) <= 0){

    return FALSE;
}
else {

    return TRUE;
}

}

function upload_file($user_id, $data, $type){

if($type == "mixes"){

$typelink="mix";

}
else {

$typelink="sample";
}

if(file_type_is_allowed($data['userfile']['type'])){

    if(file_exist($data['userfile']['name'], $_POST['directories'],
        $type)){

```



```

return "File exists!";
}

else (

$dirname = get_dir_name($_POST['directories'], $type);

$uploaddir = 'sample_library/'.get_root_dir
($_SESSION['user_id']).'/'.$type.'/'.$dirname.'/';
$uploadfile = $uploaddir . basename(str_replace(" ", "_",
$_FILES['userfile']['name']));

if (move_uploaded_file($_FILES['userfile']['tmp_name'],
$uploadfile)) {

require ('classAudioFile.php');
$AF = new AudioFile;
$AF->loadFile($uploadfile);
$AF->getSampleInfo();
$sampling_rate = $AF->wave_framerate;
$filesize = $AF->wave_size;
$no_channels = $AF->wave_channels;
$byterate = $AF-> wave_byterate;
$resolution = $AF-> wave_bits;
$filename = str_replace(" ", "_", $_FILES['userfile']['name']);
// $filename = $AF-> wave_filename;
$length = $AF->wave_length;
$format = $AF->wave_type;
$compression = $AF->getCompression($AF->wave_compression);
$url = $uploaddir;
$comments = $_POST['comments'];
$dir_id = $_POST['directories'];
$link = connect_mysql();
$query = "INSERT INTO files SET date=CURDATE(),url='".
$url."',sampling_rate='". $sampling_rate."',filesize='".
$filesize."',no_channels='". $no_channels."',bitrate='".
$byterate."',resolution='". $resolution."',filename='".
$filename."',length='". $length."',comments='".
$comments."',dir_id='". $dir_id."',format='".
$format."',compression='". $compression."', type='". $type.'";

$result = mysql_query($query, $link) or die(mysql_error());
return 'Το αρχείο είναι έγκυρο και έχει σωθεί στην βάση
δεδομένων.<br><br>
<a href="ss.php?page=upload_'.$typelink.'">
φόρτωση νέου δείγματος</a> ';

}

```

```

        else {
            return 'Το αίτημα σας απορρίφθηκε!
                <br><br><a href="ss.php?page=upload_'. $typelink.
                "'>Παρακαλώ προσπαθείστε πάλι</a>';
        }

    }

}

else {

    return 'Λάθος τύπος αρχείου, έγκυροι τύποι αρχείων είναι
        : ogg, mp3, wav <br><br>
        <a href="ss.php?page=upload_'. $typelink. "'>
        Παρακαλώ προσπαθείστε πάλι</a>';
    }

}

function file_type_is_allowed($type){

switch ($type) {
case "audio/ogg":
    return true;
    break;
case "audio/wav":
    return true;
    break;
case "audio/aiff":
    return true;
    break;
case "audio/mpeg":
    return true;
    break;
default:
    return false;
}

}

}

```

Το πρώτο function επιτρέπει στον χρήστη να φτιάξει το δικό του φάκελο μέσα στην βάση δεδομένων αλλά και μέσα στον server. Στην συνέχεια δίνουμε το όνομα του αρχείου αλλά και το όνομα του αρχείου που σώζεται ώστε να μπορεί να ανακαλεστεί από τους υπόλοιπους χρήστες. Το URL μας δίνει το πού αυτό θα σωθεί μέσα στον server μετά κοιτάει το πρόγραμμα άμα υπάρχει άλλο αρχείο με το ίδιο όνομα μέσα στον ίδιο φάκελο. Αν υπάρχει τότε μας απαγορεύει να το σώσουμε κάτω από αυτήν την ονομασία και μας υποχρεώνει να το μετονομάσουμε. Πιο σημαντικό είναι η αποκωδικοποίηση του αρχείου από όπου σε συνδυασμό με το audioclass.PHP παίρνουμε όλες τις επιμέρους πληροφορίες του αρχείου όπως δειγματοληψία, bitrate, αριθμό καναλιών κ.τ.λ. και σώζονται σε ξεχωριστά πεδία στην βάση ώστε να μπορούν να εμφανιστούν έπειτα όταν ψάχνουμε για κάποιο δείγμα ή μίξη. Τέλος το τελευταίο

function κοιτάει αν το αρχείο είναι έγκυρο δηλαδή αν ανήκει στην κατηγορία των wav, aiff, mp3 ή ogg.

Όλα τα παραπάνω λοιπόν μέσα στο functions.PHP καλούνται από το χρήστη μέσα από το ss.PHP το οποίο χρησιμοποιεί τα functions που έχουμε σώσει μέσα στο functions.PHP. Παρακάτω βρίσκεται ο κώδικας μέσα στο ss.PHP.

```
if(isset($_POST['search_db']) || $_GET['page'] == 'search'){

    if(isset($_POST['search_db'])){
        $content = get_content('search');
        $results = search($_POST['filename']);
        $content .= $results;
        echo get_page($content, $_SESSION['bar']);
    }
    else {

        $content = get_content('search');

        echo get_page($content, $_SESSION['bar']);

    }

}

else if (isset($_GET['page']) && $_GET['page'] ==
        'sample_sharing') {

    $content .= get_content('sample_sharing');
    echo get_page($content, $_SESSION['bar']);

}

else if (isset($_GET['page']) && $_GET['page'] ==
        'browse_samples') {

    $content .= get_content('browse_samples');
    echo get_page($content, $_SESSION['bar']);

}

else if (isset($_GET['page']) && $_GET['page'] ==
        'view_samples') {

    $content .= get_db_content('view_samples',
        $_GET['user_id'], '', '');
    echo get_page($content, $_SESSION['bar']);

}
```

```

}

else if (isset($_GET['page']) && $_GET['page'] == 'view_file_info') {

    $content .= get_db_content('view_file_info',
        $_GET['user_id'], $_GET['dir_id'], $_GET['file_id']);
    echo get_page_info($content, $_SESSION['bar']);

}

else if (isset($_GET['page']) && $_GET['page'] == 'delete') {

    delete_file($_GET['dir_id'], $_GET['file_id']);
    $content .= get_db_content('view_samples_dir',
        $_GET['user_id'], $_GET['dir_id'], "");
    echo get_page($content, $_SESSION['bar']);

}

else if (isset($_GET['page']) && $_GET['page'] == 'delete_sample_dir')

    delete_dir($_GET['dir_id'], "samples");
    $content .= get_db_content('view_samples', $_GET['user_id'], '', "");
    echo get_page($content, $_SESSION['bar']);

}

else if (isset($_GET['page']) && $_GET['page'] == 'upload_sample') {

    if(isset($_POST['create_dir']))(

```

```

$dirname = clean_string($_POST['custom_dir']);

if(strlen($dirname) > 1){

    create_dir($_SESSION['user_id'], $dirname, "samples");

}
else {

    echo get_page("Error creating dir",$_SESSION['bar']);

}

}

if(isset($_POST['upload_sample'])) {

    echo get_page(upload_file($_SESSION['user_id'], $_FILES, "samples"),
$_SESSION['bar']);

}
else {
$content .= get_content('upload_sample');
echo get_page($content,$_SESSION['bar']);
}

}

```

Τα ίδια ακριβώς που ισχύουν στο «Ανταλλαγή Sample» ισχύουν και για το «Ανταλλαγή Μίξης» όπως αναφέραμε και παραπάνω με την διαφορά πως όπου γράφουμε sample στον κώδικα για τα samples γράφουμε mix στον κώδικα για τις μίξεις.

Ας δούμε τώρα τι συμβαίνει όμως όταν προσπαθούμε να βρούμε ένα δείγμα μέσα στο audile και πιο ειδικά πως καλείται η τελευταία και πιο σημαντική σελίδα που είναι αυτή των πληροφοριών ενός δείγματος ή μιας μίξης.

Εδώ βλέπουμε τον κώδικα μέσα στο functions.PHP

```

function get_file_info($file_id){

$link = connect_mysql();
$query = "SELECT * FROM files WHERE file_id='".$file_id."'";
$result = mysql_query($query, $link) or die(mysql_error());

    require ('classAudioFile.php');

$html = '';

while ($row = mysql_fetch_assoc($result)) {

$AF = new AudioFile;
$filename = "".$row['url'].$row['filename'];
    $AF->loadFile($filename);
    if ($AF->wave_id == "RIFF")
    {
        $AF->visual_width=300;
        $AF->visual_height=100;

        $AF->getVisualization(substr($filename,0,strlen
                                ($filename)-4)."png");
        $img = "<br>";

    }

$html .= '
<tr>
<td>'.$row['filename'].' &nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.$row['filesize'].' &nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.$row['bitrate'].' &nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.$row['date'].' &nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.$row['resolution'].' bits&nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.$row['no_channels'].' &nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.$row['format'].' &nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.$row['sampling_rate'].' &nbsp;&nbsp;&nbsp;&nbsp;</td>
<td>'.substr($row['length'],0,4).'s&nbsp;&nbsp;&nbsp;&nbsp;</td>

```

---

```

        <td>' . $row['comments'] . ' &nbsp; &nbsp; &nbsp; &nbsp; </td>
        <td>' . $row['compression'] . ' &nbsp; &nbsp; &nbsp; &nbsp; </td>
        <td>&nbsp; &nbsp; ' . $row['type'] . ' </td>
    </tr>
    <tr>
    <td colspan="12" align="center">' . $img . ' </td>
    </tr>

    ';

}

return $html;
}

```

Ο κώδικας αυτός είναι υπεύθυνος για την εμφάνιση των πληροφοριών αλλά και της κυματομορφής ενός δείγματος ή μιας μίξης και παίρνει κυρίως τα ορίσματά του από το `Audioclass.PHP` αρχείο. Υπάρχουν επίσης πολλά άλλα `functions` που χρησιμοποιήθηκαν προκειμένου να εμφανιστούν στις προηγούμενες σελίδες το όνομα του χρήστη, το πόσα αρχεία περιέχονται μέσα στο κάθε φάκελο κ.τ.λ. αλλά το πιο σημαντικό κομμάτι είναι αυτό, για αυτό και ασχοληθήκαμε μόνο με το συγκεκριμένο.

Πάλι όπως και προηγουμένως τα `functions` αυτά καλούνται από ένα άλλο αρχείο και αυτό είναι το `content.PHP` με τον παρακάτω τρόπο.

```

function get_db_content($page, $user_id, $dir_id,$file_id){

require_once("functions.php");

if($page == 'view_mixes' && user_id_exist($user_id)){

$name = get_user_name($_GET['user_id']);
$surname = get_user_surname($_GET['user_id']);

return'

<table class="content_2" width="650">
<tr>
<td align="center"><b>Δείτε τις μίξεις από τον χρήστη '. $name .'
'. $surname .' </b></td>
</tr>
</table>

<br>

<table>
<tr>
<td width="120"></td>
<td>

<table class="content_2" width="650">
<tr><td><u>Όνομα φακέλου</u></td><td align="center"><u>Αρχεία</u>
</td><td></td></tr>

'.get_dirs_view($user_id,"mixes").'
</table>

</td></tr></table>

```



```

';
}

else if($page == 'view_mixes_dir' && user_id_exist($user_id)
      && dir_id_exist($dir_id)){

$name = get_user_name($_GET['user_id']);
$surname = get_user_surname($_GET['user_id']);
$dirname = get_dir_name($dir_id, "mixes");
return'

<table class="content_2" width="650">
<tr>
<td align="center"><b>Φάκελος '. $dirname .' από τον χρήστη '. $name .'
                      '. $surname .' </b></td>
</tr>
</table>

<br>

<table>
<tr>
<td width="50"></td>
<td>

<table class="content_2" width="600">
<tr><td><u>Όνομα αρχείου</u></td><td align="center">
      <u>Πληροφορίες</u></td><td align="center">
      <u>Ακρόαση/Αποθήκευση</u></td></tr>
'.get_files_view($dir_id,"mixes").'
</table>

</td></tr></table>

';
}

else if($page == 'view_samples' && user_id_exist($user_id)){

$name = get_user_name($_GET['user_id']);
$surname = get_user_surname($_GET['user_id']);

return'

```

```

<table class="content_2" width="650">
<tr>
<td align="center"><b>Δείτε τα αρχεία από τον χρήστη '. $name .'
                                '. $surname .' </b></td>
</tr>
</table>

<br>

<table>
<tr>
<td width="80"></td>
<td>

<table class="content_2" width="650">
<tr><td><u>Όνομα φακέλου</u></td><td align="center">
    <u>Αρχεία</u></td><td></td></tr>

'.get_dirs_view($user_id,"samples").'
</table>
</td></tr></table>

';

}

else if($page == 'view_samples_dir' && user_id_exist($user_id)
        && dir_id_exist($dir_id)){

$name = get_user_name($_GET['user_id']);
$surname = get_user_surname($_GET['user_id']);
$dirname = get_dir_name($dir_id, "samples");
return'

<table class="content_2" width="650">
<tr>
<td align="center"><b>Φάκελος '. $dirname .' από τον χρήστη '. $name .'
                                '. $surname .' </b></td>
</tr>
</table>

<br>

<table>
<tr>
<td width="90"></td>
<td>

```

```

<table class="content_2" width="650">
<tr><td><u>Όνομα αρχείου</u></td><td align="center">
    <u>Πληροφορίες</u></td><td align="center">
    <u>Ακρόαση/Αποθήκευση</u></td></tr>
'.get_files_view($dir_id,"samples").'
</table>

</td></tr></table>

';
}

else if($page == 'view_file_info' && file_id_exist($file_id)){

$name = get_user_name($_GET['user_id']);
$surname = get_user_surname($_GET['user_id']);
$filename = get_file_name($file_id);
$url = get_file_url($file_id);
return'

<table class="content_2" width="100%">
<tr>
<td align="center"><a href="/final/' . $url . '/' . $filename . '" target=
"_blank"><b><u>Αρχείο</u>: ' . $filename . ' από τον χρήστη
' . $name . ' ' . $surname . '</b></td>
</tr>
</table>

<br>

<table class="content_2" width="100%" align="center">
<td><b>Όνομα </b></td>
<td><b>Μέγεθος (bytes)</b></td>
<td><b>bitrate</b></td>
<td><b>Ημερομηνία</b></td>
<td><b>Ανάλυση</b></td>
<td><b>Κανάλια</b></td>
<td><b>Τύπος</b></td>
<td><b>Δειγματοληψία</b></td>
<td><b>Διάρκεια</b></td>
<td><b>Σχόλια</b></td>
<td><b>Συμπίεση</b></td>
<td><b>Κατηγορία</b></td>
</tr>
'.get_file_info($file_id).'
</table>
<table align="center" width="100%">
<tr>
<td align="center" width="100%">
<a href="javascript:window.close();">Κλείσιμο παραθύρου</a>

```

```

</td>
</tr>
</table>

';

}

else{

return '
<table class="content_2">
<tr>
<td>Υπήρχε σφάλμα-Προσπαθήστε ξανά!</td>
</tr>
</table>

<br>
';

}

}

?>

```

Εδώ παρατηρούμε πως υπάρχει ένα function μέσα στο `content.PHP` το **function** `get_db_content`. Το χρησιμοποιούμε εδώ προκειμένου να μας λύσει τα χέρια και να μας εμφανίσει όλο το περιεχόμενο της βάσης μέσα στο `content` που είναι και το κύριο μέρος της ιστοσελίδας. Εκεί δηλαδή που εμφανίζονται όλες οι πληροφορίες.

Αυτά ήταν και τα πιο σημαντικά σημεία από πλευράς προγραμματισμού στο `audile`. Τα υπόλοιπα `links` δεν παρουσιάζουν τέτοιο προγραμματιστικό ενδιαφέρον και καλό θα είναι να παραληφθούν.

## 4.5 AudileRadio

Πριν από την ανάλυση του ραδιοφώνου του `Audile` πρέπει να γίνει μια μικρή εισαγωγή στο `Streaming` και τα `Streaming Media`.

### 4.5.1 Streaming Media

Ο όρος `streaming media`, αναφέρεται σε πληροφορία οπτική ή ακουστική η οποία έχει τη δυνατότητα να αναπαράγεται κατά τη διάρκεια της καταφόρτωσης (`download`) από το δίκτυο. Στην πραγματικότητα οι τεχνολογία `streaming` βασίζεται στην ύπαρξη μίας προσωρινής μνήμης (`buffer`) όπου αποθηκεύονται προσωρινά τα δεδομένα καθώς καταφορτώνονται από το δίκτυο, και μέχρι να αναπαραχθούν. Μετά την αναπαραγωγή τους στον παραλήπτη, η μνήμη αυτή είναι έτοιμη να φιλοξενήσει το νέο ρεύμα δεδομένων που καταφθάνει από το δίκτυο.

### Ιστορικά:

Οι προσπάθειες να επιδειχθούν τα media στους υπολογιστές χρονολογούνται από τις αρχές των υπολογιστών, στα μέσα του 20ού αιώνα. Εντούτοις, λίγη πρόοδος σημειώθηκε για αρκετές δεκαετίες, υφιστάμενη πρώτιστα λόγω του υψηλού κόστους και έπειτα τις περιορισμένες ικανότητες των υπολογιστών.

Τα ακαδημαϊκά πειράματα στη δεκαετία του '70 απέδειξαν τις βασικές έννοιες και τη δυνατότητα πραγματοποίησης του streaming στους υπολογιστές. Κατά τη διάρκεια της πρόσφατης δεκαετίας του '80, οι υπολογιστές έγιναν αρκετά ισχυροί να επιδείξουν τα διάφορα media.

Τα αρχικά τεχνικά ζητήματα με τη ροή ήταν:

1. Κατοχή δυνατής CPU και bus-bandwidth ώστε να υποστηρίξει τα απαραίτητα data rates.
2. Η δημιουργία καναλιών με πολύ μικρό χρόνο διακοπής στο λειτουργικό σύστημα ώστε να αποτραπεί το buffer underrun. Το buffer underrun ή αλλιώς buffer underflow είναι μια κατάσταση όπου το buffer που χρησιμοποιείται για να επικοινωνήσουν οι δύο συσκευές ή οι δύο επεξεργαστές έχει γεμίσει με δεδομένα μικρότερης ταχύτητας από αυτά τα δεδομένα που εξέρχονται από αυτό το buffer. Αυτό έχει σαν αποτέλεσμα το πρόγραμμα ή η συσκευή που διαβάζει από αυτό το buffer να σταματήσει προκειμένου να γεμίσει ξανά κάτι που μπορεί να προκαλέσει προβλήματα αφού τα δεδομένα που μπαίνουν στο buffer δεν είναι κατάλληλα για διακοπή και επανεκκίνηση της ροής της πληροφορίας

Εντούτοις τα δίκτυα των υπολογιστών ήταν ακόμα περιορισμένα και τα media μεταδίδονταν πιο εύκολα από μέσα που δεν είναι streaming όπως π.χ. τα CD-ROM

Η πρόσφατη δεκαετία του '90 είχε να αποδείξει:

1. μεγαλύτερο εύρος ζώνης (bandwidth) στα δίκτυα
1. αυξανόμενη πρόσβαση στα δίκτυα, ειδικά το διαδίκτυο
2. χρήση των τυποποιημένων πρωτοκόλλων όπως το TCP/i IP, το HTTP και το HTML
3. Η εμπορευματοποίηση του διαδικτύου

Αυτές οι πρόοδοι στη δικτύωση υπολογιστών σε συνδυασμό με τους ισχυρούς προσωπικούς υπολογιστές και τα σύγχρονα λειτουργικά συστήματα συνέβαλαν ώστε να γίνουν τα Streaming Media πρακτικά και προσιτά για τους συνηθισμένους καταναλωτές. Γενικά το περιεχόμενο των πολυμέσων (Multimedia) είναι μεγάλο, έτσι οι δαπάνες αποθηκευτικών μέσων και μέσων μετάδοσης είναι ακόμα σημαντικές για να αντισταθμίσουν αυτό που γίνεται είναι τα μέσα να συμπιέζονται και για την αποθήκευση αλλά και για τη ροή. Ένα media stream μπορεί να είναι είτε on demand είτε live. On demand τα streams αποθηκεύονται σε έναν κεντρικό υπολογιστή για μια μακριά χρονική περίοδο, και είναι διαθέσιμα για να διαβιβαστούν μετά από αίτημα ενός χρήστη. Τα live streams είναι διαθέσιμα μόνο σε έναν ιδιαίτερο χρόνο, όπως σε ένα τηλεοπτικό (video) stream ενός ζωντανού αθλητικού γεγονότος ή ενός Internet ραδιοφώνου όπως το ραδιόφωνο του Audile.

#### 4.5.2 Αποθήκευση Streaming και Bandwidth

Αυτό που χρειάζεται ένας server προκειμένου να μπορεί να φιλοξενήσει ένα ραδιόφωνο είναι μια γρήγορη σύνδεση στο internet. Ας υποθέσουμε λοιπόν πως έχουμε μια γραμμή ADSL στα 256kbps. Σε αυτή την περίπτωση ας πούμε πως θέλουμε να εκπέμπουμε στα 24kbps. Κάνοντας μια απλή διαίρεση  $256\text{kbps}/24\text{kbps}=10$  βγάζουμε τον μέγιστο αριθμό επισκεπτών που μπορούμε να φιλοξενήσουμε. Όπως καταλαβαίνουμε για την περίπτωση του ραδιοφώνου του τμήματος ένας τέτοιος αριθμός είναι πολύ μικρός οπότε η σύνδεση μας πρέπει να είναι πολύ γρηγορότερη από μια απλή ADSL.

Το μέγεθος αποθήκευσης Streaming media υπολογίζεται από το bandwidth και το μήκος media με τον ακόλουθο τύπο (για έναν χρήστη και ένα αρχείο):

*μέγεθος αποθήκευσης (MBytes) = ποσοστό δυαδικών ψηφίων · μήκος*

$$\text{size(MB)} = \left( \text{length(s)} \cdot \text{bit rate} \left( \frac{\text{kbit}}{\text{s}} \right) \right) \cdot \frac{1000 \text{ bit}}{1 \text{ kbit}} \cdot \frac{1 \text{ byte}}{8 \text{ bits}} \cdot \frac{1 \text{ MB}}{1,048,576 \text{ bytes}}$$

Παράδειγμα:

Μια ώρα του βίντεο που κωδικοποιείται σε 300 KBit/S (αυτό είναι ένα χαρακτηριστικό ευρυζωνικό βίντεο για το 2005 και κωδικοποιείται συνήθως σε ανάλυση pixel 320X240) θα είναι:

$(3.600 \text{ s} \cdot 300) \text{ KBit/S} / 8,388.608 = 128,7 \text{ MB}$  εάν το αρχείο αποθηκεύεται σε έναν κεντρικό υπολογιστή για το on-demand stream. Εάν αυτό το stream απορροφάται από 1.000 ανθρώπους ταυτόχρονα, θα χρειαζόσαστε  $300 \text{ KBit/S} \cdot 1.000 = 300.000 \text{ KBit/S} = 300 \text{ MBit/S}$  του bandwidth. Αυτό είναι ισοδύναμο με 125.68 GB ανά ώρα.

#### 4.5.3 Πρωτόκολλα Streaming Media

Με τον σχεδιασμό ενός πρωτοκόλλου δικτύων που υποστηρίζει τα streaming media προκύπτουν πολλά ζητήματα. Τα πρωτόκολλα διαγραμμάτων δεδομένων, όπως το User Datagram Protocol (UDP), στέλνουν τα streaming media ως σειρά μικρών πακέτων. Αυτό είναι απλό και αποδοτικό, εντούτοις τα πακέτα είναι εύκολο να χαθούν ή να αλλοιωθούν κατά τη μεταφορά. Ανάλογα με το πρωτόκολλο και την έκταση της απώλειας, ο client μπορεί να είναι σε θέση να ανακτήσει τα αρχεία με κάποιες τεχνικές διορθώσεως λάθους, ή μπορεί να μεταπηδήσει (Interpolation) πέρα από το ελλείπον στοιχείο, ή μπορεί να υποστεί μια απώλεια.

Το Real Time Streaming Protocol (RTSP), το Real-time Transport Protocol (RTP) και το Real-time Transport Control Protocol (RTCP) σχεδιάστηκαν συγκεκριμένα ώστε να επιτυγχάνεται το media streaming μέσα στα δίκτυα. Τα τελευταία δύο βασίζονται πάνω στο UDP. Τα αξιόπιστα πρωτόκολλα, όπως το Transmission Control Protocol (TCP), εγγυώνται τη σωστή παράδοση κάθε πακέτου του media stream. Εν τούτοις, το ολοκληρώνουν αυτό με ένα σύστημα διαλειμάτων και επανεκκίνησης, το οποίο τους καθιστά πιο σύνθετους για να εφαρμοστούν. Επίσης σημαίνει ότι όταν υπάρχει απώλεια στοιχείων στο δίκτυο, το media stream περιμένει ενώ οι χειριστές του

πρωτοκόλλου ανιχνεύουν την απώλεια και αναμεταδίδουν το ελλείπον στοιχείο. Οι clients μπορούν να ελαχιστοποιήσουν την επίδραση αυτού με την αποθήκευση των δεδομένων αυτών.

Ένα άλλο θέμα είναι το ότι τα firewalls είναι πιο επιρρεπή στο να σταματούν τα αρχεία που είναι βασισμένα σε UDP πρωτόκολλα από ότι τα δεδομένα που είναι βασισμένα στο TCP πρωτόκολλο.

Τα πρωτόκολλα Unicast στέλνουν ένα χωριστό αντίγραφο του media stream από τον server σε κάθε client. Αυτό είναι απλό, αλλά μπορεί να οδηγήσει στον ογκώδη διπλασιασμό των στοιχείων όσον αφορά το δίκτυο.

Τα Multicast πρωτόκολλα αναλαμβάνουν να στείλουν μόνο ένα αντίγραφο του media stream πέρα από οποιανδήποτε σύνδεση δικτύων. Αυτό είναι μια αποδοτικότερη χρήση της ικανότητας των δικτύων, αλλά είναι πιο σύνθετη για να εφαρμοστεί. Επιπλέον, τα multicast πρωτόκολλα πρέπει να εφαρμοστούν στους routers των δικτύων, καθώς επίσης και στους servers.

Από το 2005, οι περισσότεροι routers στο διαδίκτυο δεν υποστηρίζουν τα multicast πρωτόκολλα, και πολλά firewalls τα εμποδίζουν. Το multicast πρωτόκολλο είναι πρακτικότερο για τις οργανώσεις που τρέχουν τα δίκτυά τους, όπως τα πανεπιστήμια και οι εταιρίες. Δεδομένου ότι αγοράζουν τους routers τους και τρέχουν στις συνδέσεις δικτύων τους, μπορούν να αποφασίσουν εάν το κόστος και η προσπάθεια της υποστήριξης ενός multicast πρωτοκόλλου δικαιολογούνται από την προκύπτουσα αποταμίευση σε bandwidth.

Τα πρωτόκολλα Peer-2-peer(P2P) λειτουργούν έτσι ώστε τα media να στέλνονται από τους clients που τα ήδη κατέχουν σε clients που δεν τα έχουν και τα θέλουν. Αυτό απομονώνει τον server και τις συνδέσεις του δικτύου του αποκτώντας το δίκτυο ευχέρεια. Παρόλα αυτά, προκύπτουν τεχνικά, αποδοτικά, ποιοτικά, επιχειρησιακά και νομικά ζητήματα.

Νεότεροι camcorders μπορούν να κάνουν stream video σε έναν υπολογιστή προσπερνώντας το FireWire. Αυτό χρησιμοποιεί ένα σύστημα βασισμένο σε χρόνους κρατήσεων για να εξασφαλίσει την παραγωγή και να μπορεί να παραληφθεί από πολλούς clients αμέσως.

### **Stream and transport protocols**

- HTTP (Hypertext Transfer Protocol)

Πρωτόκολλο μετάδοσης πληροφοριών στο διαδίκτυο προκειμένου να ανταλλάσσονται πληροφορίες μεταξύ του τερματικού και του εξυπηρετητή.

- MMS (Manufacturing Message Specification)

Το MMS είναι ένα διεθνές σύστημα (ISO-9506) για μεταφορά μηνυμάτων σε πραγματικό χρόνο μεταξύ συσκευών δικτύου ή και υπολογιστικών εφαρμογών.

- RTP (Real-time Transport Protocol)

Format που χρησιμοποιείται για να παραδίδει πληροφορίες ήχου και εικόνας στο Internet.

- RTCP (Real-time Transport Control Protocol)

Πρωτόκολλο που συνοδεύει το RTP και ελέγχει την ροή της μετάδοσης των πληροφοριών αυτών που χρησιμοποιούν το RTP.

- RTSP (Real Time Streaming Protocol)

Πρωτόκολλο με εφαρμογή πάνω στα streaming μέσα και επιτρέπει στο τερματικό να ελέγχει τον εξυπηρετητή των steaming μέσων και να του δίνει εντολές όπως η αναπαραγωγή και παύση.

- RealNetworks RDT (Remote Digital Terminal)

Χρησιμοποιείται στις επικοινωνίες και κυρίως στις T1 και E1 ψηφιακές γραμμές προκειμένου να μεταφερθεί η πληροφορία στα τηλεφωνικά δίκτυα.

### **Streaming media**

Τα παρακάτω είναι προγράμματα αναπαραγωγής και λήψης streaming μέσων από και προς το Internet.

- RealPlayer

Πρόγραμμα αναπαραγωγής των streaming μέσων τύπου MP3, MPEG-4, QuickTime και Windows Media καθώς και πολλούς τύπους από RealAudio και RealVideo κωδικοποιήσεις.

- Apple Quicktime

Λογισμικό κατασκευασμένο από την Apple Computer ικανό να διαχειρίζεται πληροφορίες ψηφιακής video, ήχου, κείμενο, κινούμενα σχέδια, μουσική καθώς και πολλούς τύπους διαδραστικών πανοραμικών εικόνων (interactive panoramic images).

- FORscene

Ολοκληρωμένη πλατφόρμα για video που επιτρέπει τις μη γραμμικές (non-linear) επεμβάσεις στο video κάτι που σημαίνει πως ο χρήστης μπορεί να προσεγγίσει πολύ εύκολα κάθε καρέ ενός video.

- Macromedia Flash Player

Διανέμεται από την Adobe Systems η οποία αγόρασε την Macromedia και είναι λογισμικό για streaming διανυσματικά (vector) αλλά και raster graphics χάρη σε μια γλώσσα που ονομάζεται ActionScript καθώς επίσης και διπλής κατεύθυνσης streaming για ήχο και video.

- Microsoft Windows Media



Το Media Player πρωτοεμφανίστηκε στα Windows 3.0 και μετά στα Windows 95 και Windows NT. Από τα Windows 98 και μετονομάστηκε σε Windows Media Player.

- RealNetworks

Προμηθευτής λογισμικών για Internet μέσα. Εδρεύει στο Seattle και η εταιρία είναι ευρέως γνωστή για την δημιουργία του Real Audio που είναι συμπίεση ήχου καθώς και για το RealVideo που είναι συμπίεση video. Επίσης πολύ διαδεδομένο είναι και το RealPlayer που είναι λογισμικό αναπαραγωγής streaming μέσων.

- Shoutcast

Χρησιμοποιείται για Internet ραδιόφωνο και χρησιμοποιεί το HTTP για μετάδοση του ήχου σαν πρωτόκολλο επικοινωνίας.

- Icecast

Εξυπηρετητής επίσης για χρήσεις Internet ραδιοφώνου παρόμοιες με αυτές του Shoutcast.

- Ogg/Vorbis

Πολλές φορές καλείται και απλά Ogg. Είναι ανοιχτού τύπου κωδικοποίηση για ήχο φτιαγμένο από την Xiph.org.Foundation. Ουσιαστικά είναι μια εναλλακτική κωδικοποίηση αντί του MP3 που είναι ευρέως διαδεδομένο. Προς το παρόν πολύ χρησιμοποιείται στα παιχνίδια αφού πολλοί κατασκευαστές παιχνιδιών αποθηκεύουν τον ήχο τους με αυτή την κωδικοποίηση.

- Matroska

Οι επεκτάσεις της κωδικοποίησης είναι .mkv και .mka. Το Matroska είναι μια ιδέα ανοιχτού τύπου κωδικοποίησης πολυμέσων προκειμένου να κωδικοποιεί την πληροφορία σε μορφή αρχείου ενός συγκεκριμένου λογισμικού. Μοιάζει με το QuickTime της Apple.

#### 4.5.4 Shoutcast

Το Shoutcast είναι ένα audio streaming λογισμικό διαθέσιμο για πολλών ειδών πλατφόρμες (multiplatform) προγραμματισμένο από την Nullsoft. Την εταιρία δηλαδή που προγράμματα και το πασίγνωστο winamp. Η Nullsoft χρησιμοποιεί MP3 και AAC κωδικοποίηση και το HTTP σαν πρωτόκολλο επικοινωνίας για να εκπέμψει internet ραδιόφωνο. Το λογισμικό του server αλλά και των clients είναι διαθέσιμα για Windows, FreeBSD, GNU/Linux, Mac OS X και Solaris.

Στο κέντρο του συστήματος είναι το SHOUTcast Distributed Network Audio Server (DNAS). Το DNAS είναι υπεύθυνο για να δέχεται το σήμα που εκπέμπεται (data) από το winamp και το plug-in που του επιτρέπει τέτοιο πράγμα δηλαδή του SHOUTcast Source DSP plug-in. Στην συνέχεια το αναμεταδίδει στους χρήστες που είναι συνδεδεμένοι στο SHOUTcast DNS. Με το που εισαχθούν τα audio data από το winamp στο SHOUTcast DNS ουσιαστικά γίνεται προσιτό από οποιονδήποτε χρήστη μπει στην σελίδα του SHOUTcast. Έτσι το μόνο που απομένει είναι να αποκτηθεί ένα domain (διεύθυνση) και να καταχωρηθεί στους servers του SHOUTcast κάτι που

γίνεται εύκολα και δωρεάν μέσω του DynDNS. Το domain αντιπροσωπεύει το ραδιόφωνο μας και έτσι μπορεί να βρεθεί και να ακουστεί από τους χρήστες.

Το λογισμικό Shoutcast επιτρέπει στον χρήστη να στήσει το δικό του προσωπικό ραδιοφωνικό server προκειμένου να μπορεί να εκπέμψει Internet Radio. Το format της εξόδου του Shoutcast είναι αναγνωρίσιμο και κατανοητό από πολλαπλούς clients συμπεριλαμβανομένου και των ευρέως γνωστών Nullsoft Winamp, Windows media player, XMMS, Zinf και Apple iTunes.

### **Πρωτόκολλο Shoutcast**

#### **MP3 bit stream data**

icy-notice1:<BR>This stream requires

<ahref="http://www.winamp.com/">Winamp</a><BR>

icy-notice2:SHOUTcast Distributed Network Audio Server/Linux v1.9.2<BR>

icy-name:WLHS 89.9 FM Lakota High School

icy-genre: alternative rock

icy-url:http://wlhsradio.com

Content-Type: audio/mpeg

icy-pub:1

icy-br:48

#### **SHOUTcast tags**

icy-notice:

ένα Tag γενικότερων πληροφοριών

icy-name:

εδώ εμπεριέχεται το όνομα του καναλιού που λαμβάνουμε

icy-genre:

το είδος της μουσικής που λαμβάνουμε

icy-url:

μας δίνει το url, τη διεύθυνση δηλαδή του σταθμού

Content type:

δηλώνει το MIME τύπο δεδομένων του stream, συνήθως audio/mp3 για mp3 ραδιόφωνο. Το streaming για τηλεόραση χρησιμοποιεί το video/nsv

icy-pub:

καταδεικνύει το αν το stream είναι προσωπικό(0) ή αν είναι δημόσιο(1)

icy-br:

τέλος εδώ έχουμε το bit rate του stream σε kbit/s.

Επίσης άμα θελήσει ο χρήστης μπορεί να ενεργοποιήσει και άλλα χαρακτηριστικά του πρωτοκόλλου shoutcast όπως το icy-metaint:8000 που θα δηλώσει την τοποθεσία των πληροφοριών του κομματιού και τα δεδομένα θα εμφανίζονται ανά 8000 bytes. Το πρώτο byte των πληροφοριών είναι το μέγεθος του πακέτου αυτού των πληροφοριών. Αν κάποιος θελήσει να βρει τον ακριβή αριθμό σε bytes πρέπει αυτός ο αριθμός να πολλαπλασιαστεί επί 16.

Τέλος με τα παρακάτω:

StreamUrl='www.club977.com'

StreamTitle='singer, title'

Μπορούμε να πάρουμε το όνομα του καλλιτέχνη και τον τίτλο του τραγουδιού.

## 5 Συμπεράσματα

Αξιολογώντας το Audile θα λέγαμε πως είναι σε θέση να καλύψει τις ανάγκες του Τμήματος Μουσικής Τεχνολογίας και Ακουστικής έχοντας και τη δυνατότητα αναβάθμισης στο μέλλον. Στην παρούσα κατάσταση το πρόγραμμα είναι λειτουργικό και δεν υπολείπεται σε λειτουργίες κανενός άλλου αντίστοιχου sample sharing προγράμματος. Το μόνο στο οποίο θα μπορούσε να πει κανείς πως υστερεί το audile είναι στα γραφικά και αυτό διότι το συγκεκριμένο πρόγραμμα δεν κατασκευάστηκε για εμπορικούς λόγους αλλά για καθαρά εκπαιδευτικούς. Ο γραφικός σχεδιασμός των ιστοσελίδων (graphic design) είναι μια τέχνη η οποία στην ουσία αποσκοπεί στην εμπορικότητα των ιστοσελίδων με σκοπό να κεντρίσει το ενδιαφέρον των χρηστών και σπάνια έχει να κάνει με την λειτουργικότητα αυτή καθαυτή ενός δικτυακού τόπου. Κάτι τέτοιο στην περίπτωση του audile θα ήταν άσκοπο, χρονοβόρο και εντέλει πιθανότατα αντισυμβαλλόμενο.

Ένα άλλο θέμα που προκύπτει με την χρήση και την ανάδειξη του προγράμματος αυτού είναι το οικονομικό. Λέγοντας οικονομικό εννοούμε το σε τι βαθμό θα μπορεί ένας φοιτητής του τμήματος μουσικής τεχνολογίας και ακουστικής να βάλει στο σπίτι του μια ADSL γραμμή σύνδεσης στο Internet. Οι εποχές του ακριβού Internet στην Ελλάδα φαίνονται πάντως να αλλάζουν και όλο και περισσότεροι χρήστες σε όλη την χώρα αναβαθμίζουν τις dial-up συνδέσεις τους σε ADSL. Κάτι τέτοιο θα είναι πολύ σύντομα προσιτό και για ένα φοιτητή καθώς ήδη η τιμή μιας ADSL σύνδεσης έχει πέσει κάτω από τα 20 ευρώ το μήνα, τιμή αρκετά προσιτή.

Το Audile μπορεί να βοηθήσει καθοριστικά την ομαδικότητα των φοιτητών και επίσης να κάνει τους δεσμούς της σπουδαστικής κοινότητας ακόμα πιο ισχυρούς και συμπαγείς. Με την ανταλλαγή δειγμάτων ήχου καθώς και των μίξεών τους, οι φοιτητές θα είναι σε θέση να γνωρίσουν καλύτερα ο ένας τον άλλον σε καλλιτεχνικό επίπεδο και έτσι να δημιουργούνται καλλιτεχνικά σχήματα μέσω αυτής της δραστηριότητας.

Επίσης μέσω του ραδιοφώνου, σεμινάρια, ομιλίες και βραδιές όπως αυτές της ηλεκτροακουστικής μουσικής που διοργανώνονται στο τμήμα μας, θα μπορούν να παρακολουθούνται από κάθε φοιτητή πλέον μέσα από τον προσωπικό του υπολογιστή.

## **6 Αναφορές**

### **6.1 Βιβλιογραφία**

[1] David Sklar, DS and Adam Trachtenberg, AT. *PHP Cookbook*, California: O'Reilly & Associates, 2002, pp. 10-550.

[2] Jason Gilmore, W.J.G, *Beginning PHP5 and MySQL: From Novice to Professional*, New York: APRESS, 2004, pp. 9-712

### **6.2 Διαδικτυακές Παραπομπές**

[3] <http://freesound.iua.upf.edu/>

[4] <http://www.findsounds.com/>

[5] <http://www.apache.org>

[6] <http://www.php.net>